

# Herramienta de diseño y gestión de entrenamientos de fútbol



Grado en Ingeniería Multimedia

Trabajo Fin de Grado

Autor:  
Luis Berenguer Herrero

Tutor/es:  
José Vicente Berná Martínez

junio 2019



Universitat d'Alacant  
Universidad de Alicante



## **Herramienta de diseño y gestión de entrenamientos de fútbol**

Documento correspondiente a la memoria del Trabajo de Fin de Grado llamado “Herramienta de diseño y gestión de entrenamientos de fútbol”. Desarrollado por Luis Berenguer Herrero.

### **Versión del documento**

2019.05.20

### **Licencia**

Se permite la reproducción, distribución y comunicación pública de la obra, incluso con fines comerciales siempre y cuando reconozca y cite la obra de la forma especificada por el autor o el licenciante.



## Resumen

El presente trabajo se trata de un proyecto de definición y especificación de una aplicación tanto para web como para dispositivos móviles. Dicha aplicación consiste en un gestor de entrenamientos de fútbol destinado a entrenadores y ayudantes, donde pueden realizar diversas tareas típicas del día a día de su labor: diseñar ejercicios, preparar entrenamientos, planificar semanas de entrenamientos y ciclos de trabajo, etc.

La idea nace de un proyecto previo que consiste en un sistema web en el que un club deportivo centralizaba toda su gestión interna. Esta gestión abarcaba un gran número de tareas del día a día del club, siendo un proyecto ambicioso construido a base de requerimientos y necesidades que surgían. El principal problema de este proyecto era lo limitado que estaba en ciertas áreas concretas, como podía ser el diseño y la creación de entrenos, donde en este caso, solo permitía la subida de imágenes para reflejar el trabajo que se quería realizar en dicho ejercicio.

Estas carencias mencionadas son las que se quieren resolver con este proyecto, en el que se aplican modernas tecnologías de desarrollo y la funcionalidad se centra en resolver ese problema del diseño de ejercicios, donde los usuarios se veían obligados a recurrir a herramientas externas para completar sus tareas, muchas veces con resultados mejorables.

Por ello, esta aplicación ofrece una lista de funcionalidades muy clara: diseño de ejercicios con el diseñador de ejercicios de la herramienta, gestión de entrenamientos (incluyendo tiempos, aspectos trabajados, ejercicios realizados, etc.) y visualizaciones de históricos de trabajo de los equipos. Todo ello disponible en todas las plataformas en las que se despliega: web, iOS y Android.

## Motivación, justificación y objetivo general

En la actualidad, las nuevas tecnologías han cambiado nuestra forma de interactuar con el mundo, de resolver problemas, de interactuar con otras personas, etc. Desde realizar una simple compra sin salir de tu casa hasta hacer viajes con coches autónomos sin necesidad de conductor. Las posibilidades son infinitas, los resultados son impactantes y lo mejor es que lo serán aún más con el tiempo.

Sin embargo, lo realmente apasionante es que detrás de todos estos avances existe un componente humano, el cual se ha encargado de investigar, diseñar, desarrollar, probar e implementar cada una de estas tecnologías, en algunos casos durante años y décadas, para finalmente proporcionar un beneficio a las personas y a la sociedad.

Ese componente humano de la tecnología despertó en mí una gran curiosidad e ilusión durante los cursos del grado: Poder desarrollar software que fuera atractivo y útil para la gente. Ver las posibilidades de que una parte de nuestro día a día (periódicos digitales, aplicaciones móviles, sistemas de gestión, etc.) se desarrollara gracias a los conocimientos que íbamos adquiriendo me hizo darme cuenta de mi motivación profesional. A partir de ahí le fui poniendo más mimo e interés a cada línea de código que escribía, a cada interfaz que diseñaba, a cada función que desarrollaba, en pos de que fuera algo bonito y profesional habiendo nacido de mi propio desarrollo.

Fui a partir de ahí cuando empecé mis desarrollos personales en paralelo a los proyectos universitarios. Varias páginas webs para amigos y conocidos, pequeñas aplicaciones de prueba para ejecutar en mi móvil y bastantes cursos online para profundizar en el desarrollo web.

Sin embargo, lo más importante que surgió fue algo relacionado con el club de fútbol de mi pueblo. Allí querían llevar un control centralizado de todos los chavales y equipos del fútbol formativo del club: registrar los equipos con sus jugadores, almacenar los partidos que jugaban cada fin de semana, llevar un control de la asistencia a los entrenamientos, reflejar todos los contenidos de los entrenos de la semana para que coordinadores y directivos lo pudieran revisar, etc.

En internet existían soluciones parciales a todos estos problemas: aplicaciones para la gestión y dirección de clubs, diseñadores de entrenamientos o aplicaciones para gestionar los entrenamientos de tu propio equipo. El reto aquí era realizar una herramienta transversal que

abordara todos los niveles de organización de un club deportivo (en este caso fútbol), con la ventaja de tener unas premisas y requerimientos concretos de lo que se quería.

Tras muchas horas de práctica y de aplicar los conocimientos adquiridos, resultó una herramienta completa y funcional a la que se le da un uso diario en la actualidad. A pesar de ello, la plataforma tiene diversas funcionalidades que debido a la escasez de recursos tienen un acabado y usabilidad inferior al resto de áreas de la herramienta.

Una de las tareas más realizadas por los usuarios (en este caso entrenadores de los equipos) es el diseño de ejercicios que posteriormente se añaden a los entrenamientos de la semana, de forma que los coordinadores y gestores puedan llevar un control del trabajo de cada entrenador. La dificultad técnica que suponía crear un diseñador interactivo de ejercicios hizo que se buscaran otras soluciones. Finalmente, se optó por un diseñador de ejercicios que consistía en añadir una serie de datos como material necesario, clasificación del ejercicio, descripción, etc. junto con una imagen ilustrativa del ejercicio. Una vez se tienen los ejercicios, se procede a la creación del entreno, en la que se selecciona la fecha, título, unos comentarios y finalmente los ejercicios que se realizarán en el entreno y durante cuánto tiempo.

En la práctica, las imágenes que los entrenadores añaden a los ejercicios son capturas o fotografías de otras aplicaciones externas con las que diseñan los ejercicios para llevarlos a la plataforma. Además, en el día a día hay muchos entrenamientos que son muy parecidos y poseen partes en común (el mismo calentamiento, la misma serie de ejercicios al empezar, etc.) y el hecho de que cada nuevo entrenamiento se cree desde 0 sin la posibilidad de gestionar unas plantillas de entrenamiento o cargar información de forma predefinida hace que acabe siendo una tarea bastante engorrosa de hacer para los entrenadores.

Tras detectar este hándicap realicé una pequeña investigación sobre aplicaciones o herramientas que implementaran de forma efectiva la limitación que posee mi sistema. El resultado de la investigación fue ver muchas herramientas poco accesibles o difíciles de usar, otras realmente efectivas pero cortas de funcionalidad en móviles y tabletas y finalmente otras con unos precios inaccesibles para el usuario objetivo.

Es por ello que tiempo después de la finalización del desarrollo planteé una solución propia a dicho problema, que es la creación de un diseñador interactivo de ejercicios y entrenamientos en forma de aplicación web pensada tanto para escritorio como para móviles.

La motivación del proyecto es poder aplicar todos los conocimientos adquiridos durante el estudio del grado y explorar nuevas tecnologías y lenguajes de programación desconocidos por

mí hasta el momento y que tienen un gran peso en el mercado laboral. Además, también es importante como desarrollador trabajar la versatilidad y la rápida adaptación a los cambios, sobre todo en un sector profesional tan volátil y que rara vez te permite instalarte en tu zona de confort.

En cuanto a los resultados finales, se espera una aplicación completamente nueva, independiente y abierta al público que resuelva la necesidad de los entrenadores de fútbol de poder organizar sus sesiones de trabajo de forma cómoda y efectiva, de forma que reduzca considerablemente el tiempo dedicado a estas tareas y que tengan siempre a mano todo el trabajo planificado.

Lo que también es muy interesante de la aplicación es la versatilidad futura que puede llegar a tener: se puede comercializar como producto independiente, podría integrarse con la antigua herramienta de gestión de clubs mejorando y ampliando su funcionalidad y también puede llegar a adaptarse a otros deportes de equipo como pueden ser el baloncesto, el fútbol sala o el balonmano.

En definitiva, es una aplicación interesante de desarrollar por el aprendizaje que aporta desarrollarla, por los resultados que puede ofrecer y por la escalabilidad y crecimiento futuro que puede llegar a tener.

## Citas

*Sólo se puede crecer si estás dispuesto a sentirte incómodo y molesto al intentar algo nuevo.*

*Brian Tracy*



# Índice de contenidos

Resumen.....	3
Motivación, justificación y objetivo general .....	4
Citas.....	7
Índice de figuras .....	12
Índice de tablas .....	14
1. Introducción .....	15
2. Estudio de viabilidad .....	17
2.1. Lean Canvas.....	17
2.1.1. Segmento de clientes .....	18
2.1.2. Problema .....	18
2.1.3. Proposición de valor única .....	19
2.1.4. Solución .....	19
2.1.5. Canales .....	19
2.1.6. Flujo de ingresos.....	19
2.1.7. Estructura de costes .....	20
2.1.8. Métricas clave .....	20
2.1.9. Ventaja competitiva .....	20
2.2. Análisis de riesgos .....	21
3. Estado del arte .....	23
3.1. Antecedentes. ....	23
3.2. Estudio de la problemática.....	27
3.3. Estudio de mercado .....	27
3.3.1. Easy2Coach.....	27
3.3.2. CampusDeportivo.....	29
3.3.3. SoccerSpecific.....	30
3.3.4. SportEasy.....	32

3.3.5.	Gesdep.....	34
3.4.	Limitaciones de los sistemas .....	36
3.4.1.	Interfaces pobres.....	36
3.4.2.	Usabilidad de los diseñadores.....	37
3.4.3.	Precios y modelos de negocio de los sistemas.....	39
3.5.	Tecnologías para el desarrollo del proyecto .....	40
3.5.1.	Persistencia de datos.....	40
3.5.2.	Tecnologías Back-end.....	41
3.5.3.	Tecnologías Front-end.....	43
3.5.4.	Conclusiones y tecnologías seleccionadas .....	45
4.	Objetivos .....	47
5.	Planificación y metodología .....	48
6.	Análisis y especificación .....	50
6.1.	Introducción .....	50
6.1.1.	Propósito .....	50
6.1.2.	Ámbito del sistema.....	50
6.1.3.	Definiciones, Acrónimos y Abreviaturas .....	51
6.2.	Descripción general.....	51
6.2.1.	Perspectiva del Producto .....	51
6.2.2.	Características de los usuarios .....	51
6.2.3.	Funciones del Producto.....	52
6.2.4.	Restricciones .....	52
6.2.5.	Suposiciones y dependencias.....	53
6.2.6.	Requisitos futuros .....	53
6.3.	Requisitos Específicos .....	53
6.3.1.	Interfaces Externas.....	53
6.3.2.	Funciones .....	54
6.3.3.	Requisitos de Rendimiento .....	56

6.3.4.	Atributos del Sistema .....	57
6.3.5.	Otros requisitos .....	57
7.	Diseño.....	58
7.1.	Diseño de la persistencia.....	58
7.2.	Diseño arquitectura conceptual.....	59
7.3.	Diseño API Rest .....	60
7.4.	Diseño arquitectura tecnológica Front/Back-end .....	60
7.5.	Diseño Interacción o Experiencia de Usuario.....	62
7.6.	Diseño Interfaces.....	63
7.6.1.	Página de aterrizaje.....	63
7.6.2.	Menú principal .....	64
7.6.3.	Perfil .....	65
7.6.4.	Gestión de usuarios.....	66
7.6.5.	Diseñador de ejercicios .....	66
7.6.6.	Lista de ejercicios .....	67
7.6.7.	Creador de entrenos .....	68
7.6.8.	Lista de entrenos .....	69
7.6.9.	Lista de plantillas.....	70
7.6.10.	Mapa de navegación .....	71
7.7.	Guías de estilos.....	71
7.7.1.	Paleta cromática.....	71
7.7.2.	Tipografía.....	72
7.8.	Diseño de pruebas y validación.....	73
8.	Implementación .....	75
9.	Pruebas y validación.....	76
10.	Resultados .....	79
10.1.	Contenido demostrable .....	79
10.2.	Tiempos de desarrollo .....	79

10.3.	Contenidos de asignaturas empleados .....	80
11.	Conclusiones y trabajo futuro .....	82
	Referencias.....	83
	Apéndice I: Especificación de la API .....	85
	Apéndice II: Test de regresión de la API.....	87

# Índice de figuras

Figura 1. Lean Canvas del proyecto.....	17
Figura 2. Gestión de equipos.....	25
Figura 3. Horario y planificación de entrenamientos.....	25
Figura 4. Gestión de partidos .....	26
Figura 5. Estadísticas de los equipos .....	26
Figura 6. Menú Easy2Coach .....	28
Figura 7. Creación de ejercicios Easy2Coach.....	29
Figura 8. Menú nuevo ejercicio CampusDeportivo.....	29
Figura 9. Diseñador de ejercicios CampusDeportivo .....	30
Figura 10. Planificador de ejercicios SoccerSpecific.....	31
Figura 11. Buscador de ejercicios SoccerSpecific.....	31
Figura 12. Menú principal SportEasy .....	32
Figura 13. Creación de eventos SportEasy .....	33
Figura 14. Menú principal Gesdep .....	34
Figura 15. Gestión de equipos Gesdep .....	34
Figura 16. Pizarra táctica Gesdep .....	35
Figura 17. Limitación exportación Gesdep.....	36
Figura 18. Formulario y diseñador de ejercicios Easy2Coach .....	37
Figura 19. Diseñador pequeño de Campus Deportivo .....	38
Figura 20. Bases de datos SQL.....	45
Figura 21. Lenguaje Python.....	46
Figura 22. ReactJS.....	46
Figura 23. Sprint de trabajo actual .....	48
Figura 24. Modelo de datos de la aplicación.....	59
Figura 25. Diseño arquitectura conceptual de la aplicación. ....	60
Figura 26. Stack tecnológico de la aplicación.....	61
Figura 27. User Journey Map de la aplicación (Ruta habitual de los usuarios).....	62
Figura 28. Página de aterrizaje + apartado de login/registro. ....	64
Figura 29. Menú principal de la aplicación.....	65
Figura 30. Perfil de usuario. ....	66
Figura 31. Gestión de usuarios.....	66
Figura 32. Visualización/Gestión de ejercicios. ....	67

Figura 33. Lista y gestión de ejercicios. ....	68
Figura 34. Visualización/Gestión de entrenamientos. ....	69
Figura 35. Lista y gestión de entrenamientos. ....	69
Figura 36. Lista y gestión de plantillas.....	70
Figura 37. Mapa de navegación de la aplicación. ....	71
Figura 38. Paleta de colores de la aplicación .....	72
Figura 39. Tipografía Raleway .....	72
Figura 40. Sprint 11 de trabajo.....	75
Figura 41. Fragmento de la documentación de la API .....	79
Figura 42. Testeo de registros y autenticaciones.....	87
Figura 43. Testeo de la gestión de usuarios.....	87
Figura 44. Testeo de los métodos de ejercicios .....	88
Figura 45. Testeo de los entrenamientos y sus métodos.....	88
Figura 46. Testeo de las plantillas de entrenamientos y sus métodos .....	88

## Índice de tablas

Tabla 1. Matrices de riesgos.....	21
Tabla 2. Riesgos del proyecto.....	22
Tabla 3. Tamaños del texto .....	72
Tabla 4. Pruebas manuales del sistema .....	74

# 1. Introducción

La tecnología y el sector de la informática en concreto han provocado un cambio espectacular en todos los ámbitos de la sociedad. La informatización de las empresas, la mejora de nuestro día a día, las telecomunicaciones, etc. han hecho que haya cambiado nuestra forma de vivir e interactuar con el mundo.

Sin embargo, a pesar del gran avance, existen muchos ámbitos cotidianos en los que la tecnología no aporta todo lo que podría llegar a aportar, bien sea por la falta de interés económico por parte de empresas y desarrolladores o porque simplemente se trata de un sector muy reducido como para que existan este tipo de revoluciones.

En este último caso mencionado encontramos el ámbito de los deportes. Si bien en los últimos años el deporte de élite ha alcanzado unas cotas de perfeccionismo y excelencia (impulsadas por la tecnología) nunca antes vistas, esto no se ha visto reflejado en el deporte amateur (el cual engloba a la gran mayoría de deportistas dentro de la sociedad). Esto provoca un distanciamiento cada vez mayor entre los deportistas profesionales y amateur, en parte debido a la falta de herramientas de estos últimos.

Son miles las personas que con su labor día a día impulsan el deporte no profesional, en la mayoría de casos sin existir ningún tipo de remuneración y en caso de existir, siendo muy baja. Toda esta gente es la que promueve el deporte desde edades tempranas hasta edades adultas, muchas veces con una falta de recursos casi total.

Es por todo ello que se quiere ofrecer esta herramienta para gente de dicho sector, en concreto para entrenadores de fútbol, los cuales son los encargados de gestionar parte del desarrollo físico y deportivo de cientos de miles de niños en España.

A menudo, estos entrenadores (sobre todo con los más pequeños) deben hacer una gran cantidad de funciones y tareas para gestionar sus equipos. Desde ayudar a los niños y ejercer casi como de “maestro” hasta preparar los entrenamientos o encargarse de gestionar las altas y bajas de los chavales para que esté todo listo en el partido del fin de semana.

Sin embargo, la preparación de los entrenamientos siempre es la tarea que más tiempo consume, la más repetitiva y una de las más importantes que deben realizar. Por eso es importante que presten especial atención a esta parte, de forma que puedan llevar todos los ejercicios organizados y reducir al máximo el descontrol o la improvisación a la hora de ejecutar los entrenamientos del equipo.



A grandes rasgos, lo que la aplicación quiere ofrecer se resume en los siguientes apartados: diseñador de ejercicios, diseñador de entrenamientos (juntando y combinando ejercicios), plantillas de entrenamiento (donde ofrecer una serie de ejercicios predeterminados con los que poder basarte y empezar a trabajar) e histórico de entrenamientos y ejercicios (para poder revisar y tener a mano todo el trabajo hecho).

En definitiva, lo que pretende la aplicación es mejorar la forma en la que se organiza y planifica el trabajo de los entrenadores de fútbol, así como reducir el tiempo empleado en diseñar y plasmar sus ideas.

Por otra parte, es importante señalar que para que la utilidad de la aplicación sea completa, todo esto debe apoyarse sobre un diseño basado en la movilidad y en la posibilidad de uso en móviles y tabletas. Esto es así por dos motivos:

- Primero y muy importante, por la importancia que han adquirido estos dispositivos en el usuario y consumidor medio, desbancando hace tiempo a los ordenadores como dispositivos tecnológicos más usados para navegar por internet.
- Segundo y también muy importante, para poder obtener todo el contenido de la aplicación en cualquier sitio y poder hacer uso de él en cualquier contexto: a pie de campo, en charlas o explicaciones previas a entrenamientos, en conversaciones con otros entrenadores, para explicar un ejercicio en concreto a un ayudante o a un futbolista, etc.

Para ello, van a explorarse las principales vías de desarrollo multiplataforma del mercado, y de esta forma poder llevar a cabo un desarrollo eficiente. Para conseguirlo, siempre se apostará por la optimización de los recursos disponibles: se aprovecharán las virtudes de cada tecnología escogida, se estudiarán las posibles estructuras tecnológicas que conformarán el proyecto y se maximizará en la medida de lo posible la reutilización de código.

## 2. Estudio de viabilidad

En esta sección analizaremos en detalle la viabilidad de nuestra aplicación deportiva, exploraremos nuestras pretensiones y la solución que ofrecemos al mercado. En nuestro caso, realizaremos un estudio a través de la herramienta Lean Canvas, la cual nos permitirá visualizar y analizar el producto que vamos a ofrecer, los problemas que planeamos resolver, a quién va dirigido, qué gastos nos va a producir, a través de qué medios se van a generar ingresos, etc.

Como resultado, obtendremos un análisis de nuestra aplicación que nos ayudará a tener una visión global del proyecto mucho más concisa y determinada.

Por otra parte, estudiaremos los riesgos que se puedan presentar durante del desarrollo del proyecto. Analizaremos el grado de impacto de cada uno de ellos, así como la probabilidad de que ocurran. Además, se especificarán los planes de contingencia previstos en caso de producirse alguna de las incidencias consideradas.

### 2.1. Lean Canvas

Nuestra aplicación pretende ofrecer un producto para un sector concreto del mundo del fútbol, en concreto a todos los entrenadores, coordinadores o ayudantes de estos. Para ayudar a definir esta idea se ha desarrollado el Lean Canvas de la **¡Error! No se encuentra el origen de la referencia.**, que nos ayuda a tener una visión de nuestro proyecto con una perspectiva de modelo de negocio innovador y emprendedor.

<b>PROBLEM</b> <small>List your top 1-3 problems.</small> <div>Falta de tiempo para preparar entrenamientos</div> <div>Falta de control sobre el trabajo realizado</div>	<b>SOLUTION</b> <small>Outline a possible solution for each problem.</small> <div>Herramienta visual multiplataforma para crear ejercicios y entrenamientos de forma rápida</div> <div>Historial y registro del trabajo llevado a cabo a lo largo del tiempo</div>	<b>UNIQUE VALUE PROPOSITION</b> <small>Single, clear, compelling message that states why you are different and worth paying attention.</small> <div>Preparar y llevar un control de los ejercicios y entrenamientos realizados en diferentes dispositivos de una forma sencilla y rápida. De esta forma conseguir objetivos como el uso de la información en cualquier contexto o poder reportar o informar del trabajo que se hace a otras personas.</div>	<b>UNFAIR ADVANTAGE</b> <small>Something that cannot easily be bought or copied.</small> <div>Movilidad gracias al aspecto multiplataforma de la aplicación</div>	<b>CUSTOMER SEGMENTS</b> <small>List your target customers and sales.</small> <div>Entrenadores de fútbol no profesionales</div> <div>Entrenadores polifacéticos (realizan muchas más tareas que las que debería realizar un entrenador)</div>
<b>EXISTING ALTERNATIVES</b> <small>List how these problems are solved today.</small> <div>-Improvisación</div> <div>-Usar cualquier hoja o papel volátil</div>	<b>KEY METRICS</b> <small>List the key numbers that tell you how your business is doing.</small> <div>Tráfico de la web</div> <div>Índice de captación de usuarios</div>		<b>CHANNELS</b> <small>List your path to customers (inbound or outbound).</small> <div>Demostraciones y recomendaciones a la gente del sector a través del boca a boca</div> <div>Página web personal</div> <div>Referencia en otros productos propios relacionados</div>	<b>EARLY ADOPTERS</b> <small>List the characteristics of your ideal customers.</small> <div>Entrenadores de fútbol jóvenes (20-35 años) no profesionales con curiosidad e interés por la tecnología y las aplicaciones</div>
<b>COST STRUCTURE</b> <small>List your fixed and variable costs.</small> <div>Servidor de hosting</div> <div>Desarrollo de la aplicación</div>		<b>REVENUE STREAMS</b> <small>List your sources of revenue.</small> <div>Publicidad online AdSense de Google</div> <div>Pago premium para eliminar la publicidad y mantener registros (con sus estadísticas, etc.) ilimitadamente en el tiempo</div>		

*Figura 1. Lean Canvas del proyecto  
(Fuente propia)*

En el lienzo podemos ver un pequeño esbozo de los diferentes aspectos que definen el concepto de nuestra aplicación. Vamos a entrar en detalle para analizar y justificar las ideas mostradas en él.

#### 2.1.1. Segmento de clientes

Nuestra aplicación tiene un target muy definido. Se trata de entrenadores de fútbol no profesionales, que compaginan sus labores deportivas con su trabajo, su vida personal u otros hobbies. Además, estos entrenadores tienen una gran motivación por hacer las cosas bien en lo deportivo, ya sea por sus ambiciones y sus ganas de crecer o simplemente para ayudar o mejorar a la gente o deportistas con los que trabaje.

Dentro de este tipo de entrenadores, también tenemos los llamados “entrenadores polifacéticos”, que son los que, a la falta de tiempo que tienen, se les suma la falta de recursos con la que suelen trabajar (llevan equipos ellos solos, se tienen que encargar de otras tareas administrativas, hacer gestiones para tener sitio donde entrenar, etc.).

En una herramienta como la nuestra enfocada a la rapidez y sencillez de uso, estos se presentan como los principales consumidores de la aplicación.

##### 2.1.1.1. *Early adopters*

Se define como Early adopters aquel segmento de usuarios que apostarán en primera instancia por nuestro producto, ya sea porque lo vean algo novedoso o porque les cree curiosidad lo que ofrece y lo que pueden llegar a hacer con la herramienta.

Nuestros Early adopters serán este tipo de entrenadores no profesionales con un perfil de edad joven (20-35 años), los cuales están mucho más familiarizados con la tecnología y son más conscientes de lo que les puede ofrecer. Habitualmente, estos usuarios rehúsan de métodos clásicos como el típico “boli y papel” para organizar sus entrenos y les parece más interesante cualquier método tecnológico.

#### 2.1.2. Problema

El principal problema al que se enfrenta nuestro target de usuarios es a la falta de tiempo que tienen para preparar y organizar sus entrenamientos. Esto deriva en mayor o menor medida en un descontrol muy grande sobre el trabajo que se realiza, además de recurrir frecuentemente a la improvisación.

Además de la improvisación, algo también muy habitual es el uso de un papel o una hoja para preparar ejercicios y entrenamientos de forma rápida. El problema de esto es que el trabajo seguramente se pierda con el tiempo, eliminando así cualquier posibilidad de controlar u obtener un histórico del trabajo realizado.

#### 2.1.3. Proposición de valor única

Nuestra aplicación ofrece una forma sencilla de controlar o preparar los ejercicios y entrenamientos. Estos ejercicios y entrenamientos serán accesibles desde cualquier lugar. En definitiva, gracias a esto se lleva un control más completo, se pueden generar históricos y se puede reportar fácilmente a otras personas del trabajo realizado, ya sea a coordinadores, otros compañeros, etc.

#### 2.1.4. Solución

Para ofrecer todas estas características, se desarrolla una aplicación multiplataforma (web y dispositivos móviles) que incluye una herramienta visual para poder diseñar tus ejercicios y posteriormente almacenarlos, organizarlos, crear entrenamientos con ellos, etc.

Además de la herramienta visual, también se ofrece una gestión temporal del trabajo realizado (en forma de calendarios o históricos) para no perder información y tener organizada toda la planificación.

#### 2.1.5. Canales

La forma principal de llegar a los consumidores será a través de demostraciones presenciales o recomendaciones a la gente del sector. Esto es posible gracias a la cercanía y buena relación personal con dichas personas.

Además, a través de una página web personal se publicitará la herramienta, intentando posicionarla bien en Internet. El objetivo de esto es poder llegar a todas aquellas personas que busquen alguna solución como la que se propone.

La última forma de llegar a la gente es a través de referenciar la aplicación desde otros productos propios que tienen cierta relación con el proyecto. Principalmente, porque la aplicación mejora ciertos aspectos de este producto anterior y puede ser un salto interesante para la gente que lo utiliza.

#### 2.1.6. Flujo de ingresos

El origen divulgativo de este proyecto hace que obtener un alto beneficio económico no sea la principal motivación. Aun así, para potenciar los ingresos del proyecto se crearán dos versiones

de la aplicación: una gratuita y otra de pago. Ambas aplicaciones potenciarán las dos vías principales de ingresos: los anuncios en la versión gratuita de la aplicación y los pagos para obtener la versión completa. Dicho de otra manera, nuestra aplicación adoptará el modelo “freemium” de negocio.

La versión gratuita de la aplicación contendrá anuncios y ligeras limitaciones en cuanto a funcionalidad.

Por otro lado, la versión de pago eliminará cualquier tipo de anuncio y permitirá explotar toda la funcionalidad.

#### 2.1.7. Estructura de costes

Al ser un proyecto personal de un solo integrante, los costes de desarrollo y mantenimiento de la aplicación consisten en el servidor de hosting que aloje la aplicación y en los recursos de desarrollo invertidos por mi parte durante el proyecto. Se hará de uso de software libre y de tecnologías de código abierto, por lo que la tecnología y las herramientas a utilizar no supondrán un gasto extra en el desarrollo.

#### 2.1.8. Métricas clave

Los indicadores que nos dirán como está funcionando nuestra aplicación se basan en el tráfico e interacción de los usuarios:

- El tráfico dentro de la web será un buen medidor para saber cuan visible estamos haciendo nuestra herramienta al mundo.
- El número de usuarios dados de alta frente al tráfico que obtenemos nos dirá si nuestra herramienta es atractiva de cara al público. Igualmente, el número de descargas de la aplicación será un buen indicador de este dato.
- Por último, el porcentaje de usuarios activos dentro de la aplicación nos mostrará si realmente a nuestro público le aporta valor el uso de la herramienta.

#### 2.1.9. Ventaja competitiva

Para finalizar, la ventaja que más valor le aporta a nuestra aplicación es el aspecto móvil de esta. Al ser una aplicación multiplataforma, puede ser usada fácilmente en diferentes ámbitos y contextos. Esto es algo normal en otro tipo de aplicaciones, sin embargo, esta característica no está lo suficientemente explotada dentro de este mundillo.

## 2.2. Análisis de riesgos

Para analizar los riesgos de nuestra aplicación, hemos construido dos matrices que nos ayuden a determinar la importancia y valoración de los riesgos que se pueden presentar durante el desarrollo del proyecto. Dichas matrices las podemos encontrar en la siguiente tabla:

		RIESGO INICIAL				
		Impacto				
		Insignificante	Pequeño	Moderado	Grande	Catastrófico
Probabilidad	Muy baja	Bajo	Bajo	Bajo	Medio	Medio
	Baja	Bajo	Medio	Medio	Alto	Alto
	Media	Medio	Medio	Alto	Alto	Muy alto
	Alta	Medio	Alto	Alto	Muy alto	Extremo
	Muy alta	Alto	Alto	Muy alto	Muy alto	Extremo
		RIESGO RESULTANTE				
		Efectividad del plan de acción				
		No satisfactoria	Pequeña	Moderada	Satisfactoria	Grande
Riesgo inicial	Bajo	Bajo	Bajo	Bajo	Bajo	Bajo
	Medio	Medio	Medio	Medio	Medio	Bajo
	Alto	Alto	Alto	Alto	Medio	Medio
	Muy alto	Muy alto	Muy alto	Alto	Alto	Medio
	Extremo	Extremo	Extremo	Muy alto	Muy alto	Alto

Tabla 1. Matrices de riesgos

Gracias a esta matriz podemos obtener una ponderación final de cada riesgo definiendo tres valores:

1. Primero, la probabilidad de que se dé una circunstancia o riesgo determinado, la cual puede tener los valores: Muy baja, Baja, Media, Alta y Muy alta.
2. Segundo, el impacto que tiene la ocurrencia del riesgo en el desarrollo del proyecto. El impacto puede tener los valores: Insignificante, Pequeño, Moderado, Grande o Catastrófico. Una vez elegidos la probabilidad y el impacto obtenemos un valor de riesgo inicial definido por la primera matriz (por ejemplo, para una probabilidad Alta y un impacto Insignificante, el riesgo obtenido es Medio).
3. Por último, y una vez definido el plan de contingencia, añadiremos uno de los siguientes valores que determinan la efectividad del plan: No satisfactoria, Pequeña, Moderada, Satisfactoria y Grande. Con la segunda matriz de riesgos cruzaremos el valor de riesgo inicial con la efectividad del plan de contingencia y obtendremos un riesgo resultante representativo de la importancia real del riesgo (por ejemplo, para un riesgo inicial Alto y una efectividad Grande del plan de contingencia, el riesgo resultante será Medio, ya

que a pesar de que es probable que se dé dicho riesgo, su solución es realmente satisfactoria y seguramente no afectará al desarrollo o al producto final).

Una vez definidos los comportamientos de los valores de la matriz de riesgos, representamos los riesgos del proyecto en la siguiente tabla, donde las columnas en gris son los valores auto calculados por las matrices de riesgos:

Descripción	Prob.	Impacto	Riesgo inicial	Plan de contingencia	Efectividad del plan	Riesgo result.
Limitaciones de las tecnologías de aplicaciones "Híbridas" para trasladar una aplicación web a una aplicación móvil	Media	Moderado	Alto	Usar tecnologías para generar la aplicación móvil a partir de la aplicación web y gracias al diseño web responsive crearíamos así una aplicación adaptada a todos los dispositivos	Satisfactoria	Medio
Limitaciones en las librerías de gestos que afecten en el diseño de los ejercicios	Baja	Moderado	Medio	Llegar a una situación de compromiso mediante alternativas que satisfagan las necesidades y no afecten a la funcionalidad (P. ej: sustituir algunos gestos por clicks)	Moderada	Medio
Cambio de residencia durante el desarrollo del proyecto	Media	Moderado	Alto	Continuar el desarrollo de forma remota reduciendo la presencialidad a los momentos más necesarios	Satisfactoria	Medio
Problemas de despliegue de la infraestructura de la aplicación	Media	Grande	Alto	Desplegar la aplicación a través de servidores locales	Moderada	Alto

*Tabla 2. Riesgos del proyecto*

### 3. Estado del arte

Como hemos comentado en puntos anteriores, la revolución digital ha supuesto un enorme avance en muchos ámbitos de la sociedad. Sin embargo, el camino por recorrer aun es largo y existen ciertos sectores no muy explotados donde el potencial de mejora es bastante alto. Un caso concreto podría ser el ámbito de nuestra aplicación, el mundo del fútbol.

Existen numerosos avances que han cambiado (y deshumanizado) en parte el fútbol profesional. Hablamos de pulsómetros para controlar en todo momento la frecuencia cardíaca y el grado de esfuerzo de los futbolistas, o monitores GPS con los que hacer un seguimiento y controlar las distancias que recorre cada jugador en un entrenamiento o partido, además de las herramientas y sistemas personalizados que poseen para gestionar y centralizar toda su información.

Como en todo, el problema de esto es que ha supuesto un distanciamiento enorme entre el fútbol profesional y fútbol aficionado/fútbol base. A pesar de ello, para este último sector han ido apareciendo diversas herramientas y aplicaciones, que analizaremos en próximos puntos, que también suponen avances para el fútbol más modesto.

Sin embargo, la falta de variedad en el abanico de opciones, la falta de profundidad en sus conceptos, las limitaciones de las herramientas y la escasa comunidad de desarrolladores detrás de este sector hacen que en muchas ocasiones las necesidades de estos usuarios no se satisfagan de forma total. Ese es el punto de inicio a partir del cual surgió un interesante proyecto, el cual se describe en el siguiente apartado.

#### 3.1. Antecedentes.

Casi dos años atrás, el equipo de fútbol de la población de Novelda estaba llevando a cabo una reestructuración de su organigrama y se querían llevar a cabo una serie de cambios que facilitaran el control de los equipos y sus jugadores, así como el trabajo que realizaba cada entrenador con sus respectivos equipos.

Algunos de los requerimientos que buscaban eran los siguientes: tener un repositorio centralizado donde gestionar todos sus jugadores y equipos, poder diseñar y almacenar ejercicios de entrenamiento de forma que se fuera construyendo gradualmente una base de datos de ejercicios potente con la aportación de todos los entrenadores, poder llevar las estadísticas de los jugadores y equipos en cada partido (goles, minutos jugados, etc.) o crear y



guardar entrenamientos para que los responsables de los entrenadores pudieran visualizar fácilmente el trabajo que están realizando.

Para conseguirlo, realizaron una búsqueda por las herramientas que existían en el mercado. Lo que encontraron fue herramientas de diversos tipos, cada una de las cuales les ofrecía algún punto en concreto de lo que querían. El problema era que no existía ninguna aplicación que abordara su problema de forma completa: algunas eran muy específicas para una tarea en concreto; otras ayudaban en la gestión general de los clubes, pero no entraban en detalle en la preparación de ejercicios y entrenamientos y finalmente otras tenían un precio por usuario totalmente desorbitado.

Además de los problemas mencionados, existía el inconveniente de que la gran mayoría de sistemas ofrecían un pago por suscripción, por lo que en el momento no se renovara y no siguieran con la herramienta perderían toda la gestión realizada hasta el momento y perderían su base de datos de ejercicios.

Ante esta situación, y debido a mi relación con los integrantes de este club, se me propuso la creación de una herramienta personalizada, que abarcara todos los requisitos que fueran posibles y así tener el repositorio centralizado de información que deseaban.

Durante unos 6 meses se fue desarrollando la herramienta con los conocimientos que poseíamos en ese momento. Empezó por crearse una especie de mockup de la aplicación concreta. Es decir, se creó una versión temprana con interfaces de ejemplo, la cual poseía casi la totalidad de las interfaces que iban a desarrollarse, pero todavía sin funcionalidad ninguna.

Se presentó al club esta versión temprana de la aplicación y les gustó, por lo que empezó a desarrollarse gradualmente la funcionalidad de cada una de las interfaces, haciendo disponibles poco a poco más y más funcionalidades de la herramienta.

De la versión base, se tuvieron que ampliar algunos apartados concretos. También se tuvo que recortar algunos apartados por restricciones técnicas de desarrollo. Finalmente se obtuvo una aplicación bastante completa, con funcionalidades útiles y algunas muy interesantes:

- Gestión de jugadores del club, que es una de las funcionalidades básicas del sistema. Al dar de alta a un jugador pueden introducir su información básica, así como sus datos médicos, historial de lesiones u otra información de interés.
- Gestión de equipos del club, que es digamos el Core de la aplicación ya que el resto de funcionalidades giran en torno a ello. Los administradores pueden crear y modificar equipos, así como configurar varias cosas que veremos a continuación.



Figura 2. Gestión de equipos  
(Fuente <http://www.noveldacfgestion.com>)

- Gestión de entrenamientos de cada equipo, otra piedra angular del sistema. A cada equipo se le pueden definir una serie de días de entrenamiento a la semana. Una vez están definidos, se pueden crear ejercicios y entrenos a esos equipos en el día de la semana especificado, rellenando un horario donde los entrenamientos en verde son los que ya han sido planificados. Además, la celda del entrenamiento en verde será un link a los detalles del entreno planificado:

Semana del 02/12 al 08/12 de 2018					
Lunes	Martes	Miércoles	Jueves	Viernes	
19:30-21:00h Cadete 'A' Magdalena	17:30-18:45h Alevín 'E' Campitos	18:45-20:00h Alevín 'A' Campitos	18:45-20:00h Infantil 'A' Campitos	15:00-17:15h Infantil 'A' Campitos	
19:30-21:00h Cadete 'B' Magdalena	18:30-19:45h Infantil 'A' Magdalena	18:45-20:00h Alevín 'D' Campitos	19:30-21:00h Juvenil 'A' Magdalena	17:30-18:45h Benjamín 'A' Campitos	
	18:45-20:00h Alevín 'B' Campitos	18:45-20:00h Alevín 'C' Campitos		17:30-18:45h Alevín 'B' Campitos	
	18:45-20:00h Benjamín 'A' Campitos	19:30-21:00h Cadete 'B' Magdalena		17:30-18:45h Benjamín 'B' Campitos	
	18:45-20:00h Benjamín 'B' Campitos	19:30-21:00h Cadete 'A' Magdalena		18:45-20:00h Alevín 'A' Campitos	
	21:00-22:30h Juvenil 'A' Magdalena	21:00-22:30h Senior 'A' Magdalena		18:45-20:00h Alevín 'C' Campitos	
	21:00-22:30h Senior 'A' Magdalena			18:45-20:00h Alevín 'E' Campitos	

Figura 3. Horario y planificación de entrenamientos  
(Fuente <http://www.noveldacfgestion.com>)

- Gestión de los partidos de los equipos, donde cada equipo podrá registrar los partidos jugados cada fin de semana introduciendo todo tipo de información: fecha del partido, jornada, alineaciones, goles, minutos jugados, jugadores destacados, etc. Todo ello con una interfaz gráfica que ayude a visualizar la estrategia:



Figura 4. Gestión de partidos  
(Fuente <http://www.noveldacfgestion.com>)

- A su vez, lo que generan los partidos registrados son una serie de estadísticas globales que se pueden visualizar por cada equipo. Teniendo un historial de los minutos jugados cada partido por cada jugador, las tarjetas que han recibido los jugadores y los goles que han anotado:

2017/2018 ▼

Jugador	C	T	MT	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	J24	J25	J26	J27	J28	J29	J30	J31	J32	J33	J34
Ayala	16	13	1170		90	90	90	90	90	90	90	90						74		14		90	90	28	90		3	0	90	90	90	89					
PEPE	28	22	2092	90	90	90	90	75	25	45	90	90	90	90	90	90	90	16	90	90	90	90	90	28	90		90	45	36	67	45	90					
Carlos	22	21	1760	90		90	90	90	90	90	90	90	90	90	90	90	90	85	90		90	90	90	90	90	90	90	90	90	90	67	75					
Jaume	28	28	2471	90	90	90	90	90	90	65		90	90	90	90	90	90	77	90	90	90		90	90	90	90	90	90	90	79	90	90	90	90			
CALA	26	10	1079	90	12	45	18	25	85	90	79	90	90	90	90	90	45	13	0	26		23	16	4	45	13	24	45	8	23	20	60					
Dani	21	15	1269	16	21	16							40	90			78	90	70	74	80	57	86	90	45	78	23	79	26	30	90	90					
Imanol	28	25	2167	69	90	72	80	90	90		88	90	80	45	35	87	90	83	74	71	90	65	74	45	30	90	90	90	89	90	90	90	90				
Luismi	24	18	1437		0	0	15	15	90	45			90	90	90	90	30	15	90	64	90	90	90	90	0	45		90	90	90	38						
ALBERTO	8	1	220	1	45					0	3	87	72		0	12																					
Carra	26	20	1754	58	90	74			29	90		90	90	65	67	90	60	75	90	90	90		0	80	90	90	75	66	11	1	33	70	90				
Requena	12	3	391				24	7	22	18	55	38	24	33	50	45	55																				
MONTORO	27	27	2264	90	89		90	90	65	90	87	90	57	90		90	90	90	90	90	76	90	90	62	90	90	90	90	64	54		90	90				
JORGE	6	2	223	74					5	35	52			30	27																						
Samuel	5	1	176										33	25	23		5																				
Verdú	20	18	1680								90	89	90	90		90	90	90	90		90	33	90	32	86	90	90	90		90	90	90	90				
Rafa Juan	13	11	817																	28	19	70	45	90	45	60	67	90	86	82	45	90					
LIFANTE	4	0	42																							12		11	4			15					
Paco Leonis	1	0	0																								0										
Mané	1	0	37																																	37	

Jugador	TA	DA	TR	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15	J16	J17	J18	J19	J20	J21	J22	J23	J24	J25	J26	J27	J28	J29	J30	J31	J32	J33	J34
Ayala	7	1	0	/	/	A	A	/	/	A	A	/						A		/	/	/	/	/	/	/	/	/	/	A	A	/	DA	/	/	/	
PEPE	3	0	0	/	/	/	/	/	/	/	/	/	A	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
Carlos	4	0	0	/	/	/	/	/	/	A	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
Jaume	8	0	0	A	/	/	/	A	A	A	A	/	A	/	/	/	/	/	/	/	/	/	/	/	/	/	/	A	A	/	/	/	/	/	/	/	
CALA	7	0	0	A	/	/	/	/	/	/	/	/	A	/	/	/	/	/	/	/	/	/	/	/	/	/	/	A	/	/	/	/	/	/	/	/	
Dani	4	0	0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
Imanol	3	0	0	/	A	A	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
Luismi	6	0	1	/	/	/	/	A	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
ALBERTO	2	0	0	/	/	A	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
Carra	7	0	0	/	/	A	/	/	/	/	/	/	/	A	A	A	/	/	/	/	/	/	/	/	/	/	/	A	/	A	/	/	/	/	/	/	
Requena	1	0	0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
MONTORO	10	1	0	/	DA	/	/	A	A	A	A	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	A	/	A	/	/	/	/	/	
JORGE	0	0	0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
Samuel	0	0	0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
Verdú	2	0	0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	A	/	/	/	/	/	/	/		
Rafa Juan	4	0	0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
LIFANTE	0	0	0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
Paco Leonis	0	0	0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	
Mané	0	0	0	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	/	

Figura 5. Estadísticas de los equipos  
(Fuente <http://www.noveldacfgestion.com>)

- Además, existen otras funcionalidades menores que complementan el funcionamiento de la aplicación: registro de asistencia a los entrenamientos, seguimiento de lesiones de los jugadores, buscador de jugadores del club, buscador de ejercicios, etc.

En definitiva, resultó una herramienta que cumplía con creces los requerimientos funcionales de la gente del club. Sin embargo, es cierto que existen muchas áreas donde la capacidad de mejora en cuanto a usabilidad es grande, ya sea por la dificultad de hacer las cosas o lo repetitivo de las tareas.

### 3.2. Estudio de la problemática

Entrando en detalle, el principal problema de la herramienta era la creación y gestión de entrenos. Si bien los resultados eran buenos, los métodos no tanto. Al no tener una herramienta gráfica donde diseñar los ejercicios, se tomó la decisión de que la representación visual del ejercicio fuera una imagen, por lo que los entrenadores tienen que recurrir a herramientas externas donde diseñar sus ejercicios, sacar una imagen o una captura y subirla a la aplicación.

A pesar de que los ejercicios se pueden reutilizar en varios entrenos, este proceso se hace especialmente engorroso en el día a día, donde no es posible guardar configuraciones predeterminadas o plantillas de entrenamientos que agilicen la creación de entrenos, teniendo que rellenar siempre todo desde 0.

Pero este problema no se limita solo a nuestra aplicación. Tras la experiencia adquirida en el trato con las personas del sector y como veremos en el estudio de mercado, este problema se extiende por el resto de aplicaciones que intentan resolver problemas similares.

Ya sea por un motivo o por otro, la creación y gestión de ejercicios y entrenamientos termina siendo un problema recurrente en este mundillo, no conociendo aun personalmente ninguna herramienta que se ajuste a los requerimientos de los entrenadores de la mayoría de entrenadores.

### 3.3. Estudio de mercado

Pasamos ahora a analizar herramientas y aplicaciones del mercado cuyo principal objetivo es resolver los problemas mencionados en puntos anteriores

#### 3.3.1. Easy2Coach

Easy2Coach [1] es una aplicación pequeña y específica que está dirigida a gestionar equipos sueltos de fútbol. Para ello, permite a los entrenadores registrar información sobre el equipo y

dar de alta sus jugadores e información específica sobre ellos. Además, podemos dar de alta ejercicios o cogerlos de una base de datos compartida y así añadirlos y planificarlos en nuestros entrenamientos.

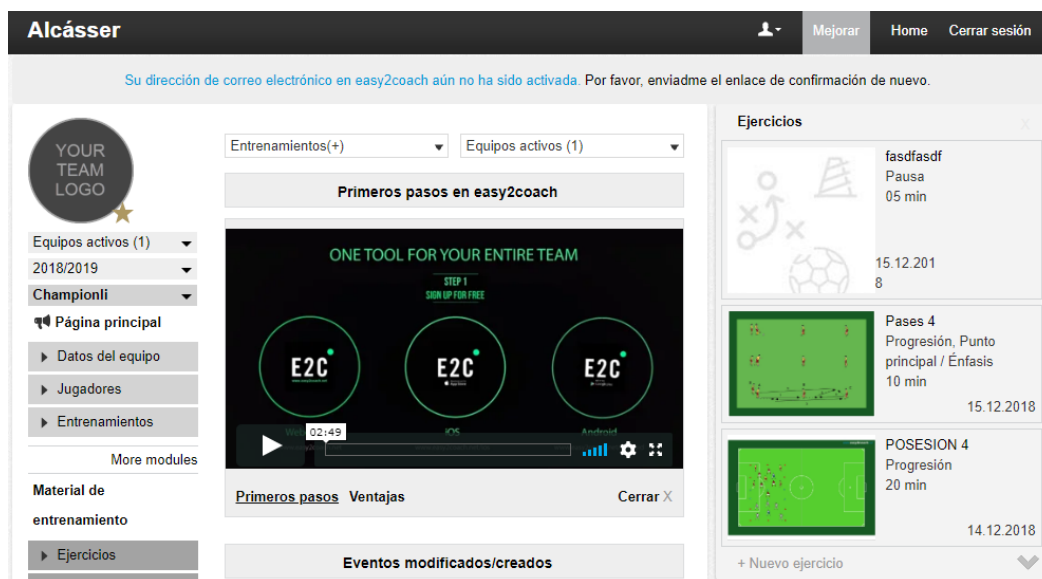


Figura 6. Menú Easy2Coach  
(Fuente <http://www.easy2coach.net>)

- Ventajas de Easy2Coach:
  - Aplicación directa, con sus funcionalidades claras y especificadas.
  - Añadir vídeos e imágenes a los ejercicios.
  - Detallar bien cada ejercicio y entrenamiento que se realiza gracias a la cantidad de información que puedes añadir.
  - Tiene aplicación descargable para iOS y Android.
- Desventajas:
  - Interfaz arcaica y difícil de usar. Muchos datos en pantalla y con errores de usabilidad (listas desplegables que no se despliegan o no se cierran, errores en los formularios que no te indican donde están, etc.).
  - No tiene un diseñador de ejercicios integrado, su solución es subir imágenes o URLs de vídeos para que tengan parte gráfica. Ofrece una opción para usar un software externo mal integrado para añadir imágenes al ejercicio.
  - Cuesta llevar la trazabilidad temporal de los ejercicios que se realizan, no apareciendo claramente en ningún timeline ni calendario.
  - En resumen, aplicación muy poco visual y poco intuitiva.

YOUR TEAM LOGO

Equipos activos (1)

2018/2019

Championli

Página principal

Datos del equipo

Jugadores

Entrenamientos

More modules

Material de entrenamiento

Ejercicios

Nuevo ejercicio

Plan de entrenamiento

Otro

Download on the App Store

GET IT ON Google play

NUEVO EJERCICIO

Ayuda

Nombre:

Idioma: Español

Estructura del entreno\*

Duración\*

Edad

Número de jugadores

Tipo de entreno

Nivel de habilidad

Campos marcados (\*) son obligatorios

Detalles

Organización:

Proceso:

Más detalles:

Youtube:

https://www.youtube.com/watch?v=

Gráficos/Archivos

Otros criterios

Haz clic para empezar Software gráfico

Haz clic para empezar Software animación

Subir archivo:

Browse...

☐ Confirmo que tengo los derechos de autor de esta imagen / archivo o que voy a usarlo únicamente de manera interna en mi equipo sin publicarlo para ningún otro usuario.

Subir imágenes / videos / archivos

El tamaño máximo permitido es de 8MB.

\* Obligatorio

Figura 7. Creación de ejercicios Easy2Coach  
(Fuente <http://www.easy2coach.net>)

### 3.3.2. CampusDeportivo

Campus Deportivo [2], a diferencia del resto de aplicaciones de la lista, es un software de escritorio que permite diseñar entrenamientos y ejercicios. Este software sirve para varios deportes de equipo: baloncesto, balonmano, fútbol sala, voleibol, futbol, etc.

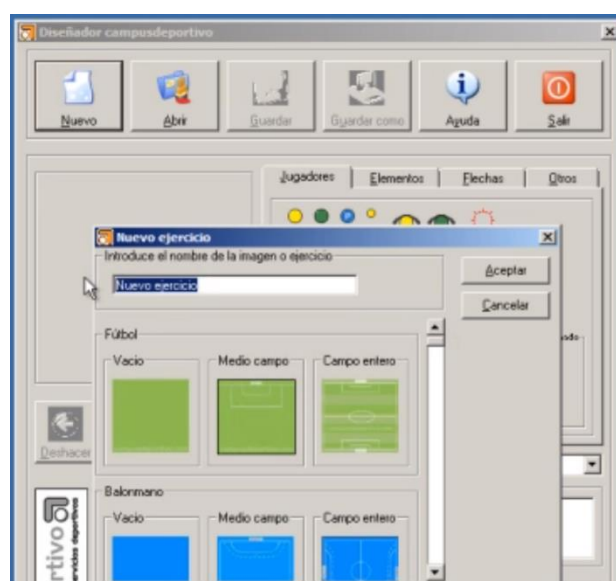


Figura 8. Menú nuevo ejercicio CampusDeportivo  
(Fuente <http://www.campusdeportivo.com>)

Las funcionalidades principales que posee es el diseño de ejercicios y entrenamientos y la exportación de estos a formatos gráficos (GIF o imágenes estáticas).

- Ventajas CampusDeportivo:
  - Variedad de deportes sobre los que preparar ejercicios y entrenamientos.
  - Exportación de ejercicios en formato gráfico para usarlos en cualquier sitio.
  - Opciones para imprimir los ejercicios y entrenos de forma estructurada.
- Desventajas:
  - Programa de escritorio que requiere instalación previa, perdiendo la comodidad de poder trabajar en cualquier sitio.
  - Animaciones simples y que son difíciles de crear.
  - Software para descargar previo pago.

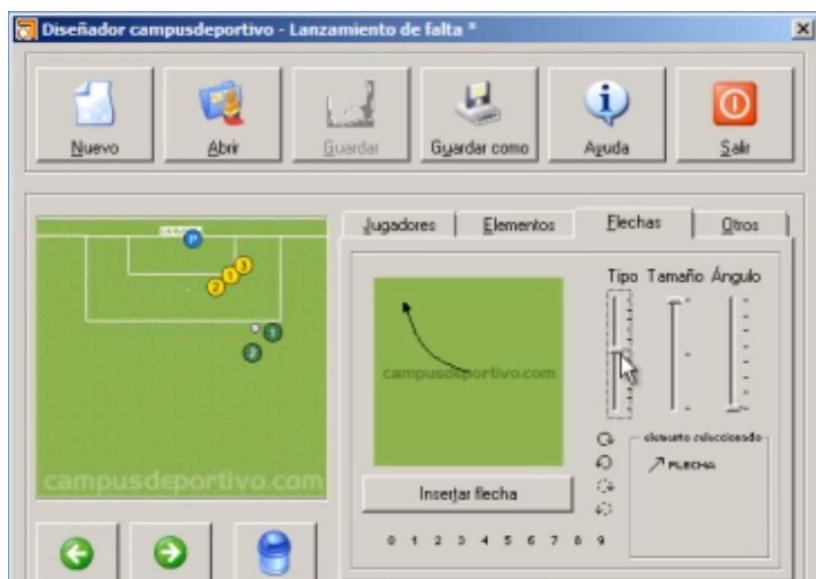


Figura 9. Diseñador de ejercicios CampusDeportivo  
(Fuente <http://www.campusdeportivo.com>)

### 3.3.3. SoccerSpecific

SoccerSpecific [3] es una plataforma online de ejercicios de fútbol. Dicha plataforma basa su funcionalidad en ofrecer una base de datos de ejercicios de todo tipo, con un avanzado buscador por filtros que permite buscar el tipo de ejercicio que queramos. Además, cada usuario puede realizar sus aportaciones subiendo sus propios ejercicios a la plataforma.

- Ventajas de SoccerSpecific:
  - Base de datos colaborativa y potente, con miles de ejercicios con muchas variantes.



- Buscador avanzado de ejercicios, en el que puedes buscar el ejercicio que se adapte a lo que necesitas.
- Diseñador de ejercicios visual y efectivo, con gran variedad de opciones y figuras.

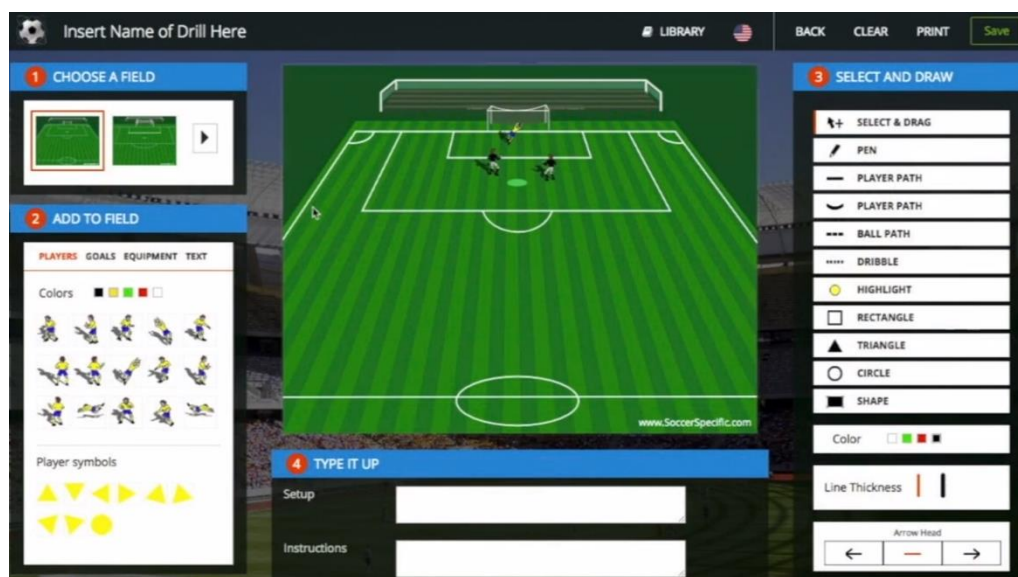


Figura 10. Planificador de ejercicios SoccerSpecific  
(Fuente <https://soccerspecific.com>)

- Desventajas:
  - Programa exclusivamente de pago. Te permite acceder a una demo de 7 días previa introducción de una tarjeta de crédito.
  - Precios poco populares (la tarifa más barata es de 5\$ mensuales) para los usuarios medios.
  - Se centra en la creación de ejercicios, no permite planificar un entrenamiento ni llevar la gestión de un equipo ni un club.

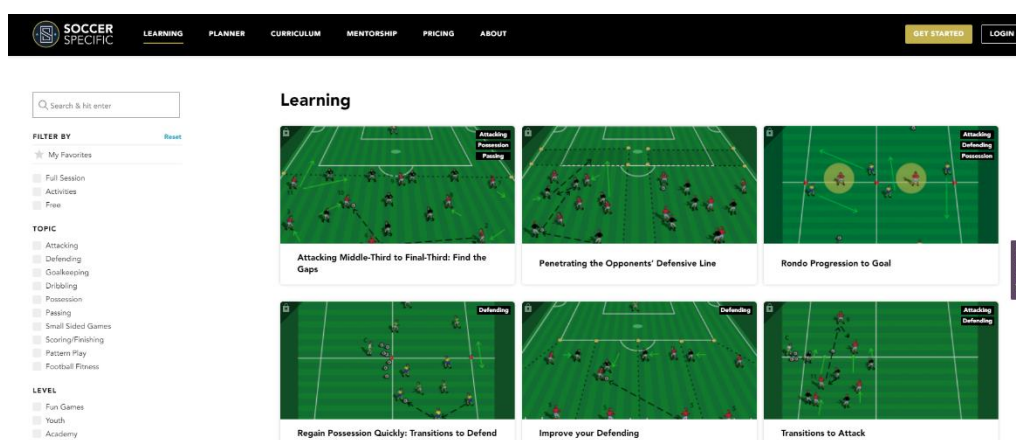


Figura 11. Buscador de ejercicios SoccerSpecific  
(Fuente <https://soccerspecific.com>)



### 3.3.4. SportEasy

Con SportEasy [4] registrar y gestionar tu equipo de casi cualquier modalidad deportiva. Para ello, te permite gestionar la plantilla del equipo, registrar asistencia de tus jugadores, ver una serie de estadísticas del equipo y de los jugadores, crear eventos como partidos amistosos, partidos de competición, torneos, etc.



*Figura 12. Menú principal SportEasy  
(Fuente <https://sporteasy.net>)*

- Ventajas de SportEasy:
  - Sencilla y eficaz gestión de tu equipo deportivo. La forma de añadir jugadores, dar de alta eventos en el calendario es efectiva y clara.
  - Posibilidad de invitar a gente a colaborar en tu equipo, de forma que el tiempo de gestión se divida y se lleve conjuntamente. Además, existen los mensajes dentro de la aplicación, con el fin de comunicarte con los colaboradores.
  - En caso de que se necesite, será posible gestionar los pagos de cuotas y mensualidades de los jugadores dentro de la aplicación.
  - Potente versión gratuita que facilita el atraer usuarios.

**Creación de eventos**

**Tipo**

Categoría \* Partido amistoso [Ayuda con las categorías](#)

Descripción Útil para agregar información detallada

Formato \* 11 contra 11 ⓘ

Adversario Adversario por determinar - [Crear un adversario](#) ⓘ

**Fecha**

Recurrencia \*   
☒ Evento único   
☐ Eventos recurrentes

Fecha/hora de inicio \* 17/12/2018 18 : 00

[Añadir una fecha/hora de fin](#)

Encuentro 18 : 00 ⓘ

Lugar del encuentro ej. frente al vestuario

*Figura 13. Creación de eventos SportEasy  
(Fuente <https://sporeasy.net>)*

- Desventajas:
  - No permite el diseño de ejercicios.
  - No se puede planificar el contenido de una sesión de entrenamiento.
  - En definitiva, no entra tanto a nivel de planificar técnicamente el trabajo de un equipo si no de tener un sitio donde tener la información del día a día del equipo resumida.
  - No parece tener ningún punto fuerte frente al resto de competidoras ya que no ofrece nada especialmente útil.

### 3.3.5. Gesdep

Gesdep [5], o Gestión Deportiva, es quizá la herramienta con más contenido y posibilidades a la hora de gestionar un club deportivo. Esta herramienta permite tanto la gestión de clubes de fútbol como de baloncesto, balonmano, fútbol sala y rugby, pero lo que se trata de una herramienta bastante trabajada.

Accesos Directos

Busqueda de personas

Últimos Mensajes

Ver	Fecha	Remitente	Asunto
	22/11/2018	Supervisor	Prueba de mensaje
	01/07/2016	Usuario 1	RE: RE: Cambio de campo
	01/07/2016	Usuario 1	Cambio de campo
	13/05/2015	Supervisor	Aviso de cambio de horario
	13/05/2015	Supervisor	Incidente en el entrenamiento
	10/05/2015	Supervisor	Reunión de coordinadores 27/05/15

¿Dudas en la creación de la nueva temporada? Mira este vídeo

Consulta el apartado de dudas frecuentes (FAQ)

Próximos Cumpleaños

20/12 Sergio (INF)	17
28/12 Manu (INF)	16
01/01 Matias (INF)	17
17/01 Fabio (JUV)	21
18/01 Manuel (CAD)	20
01/02 Julio (CAD)	20
13/02 César (JUV)	22
15/02 Rober (INF)	18

Figura 14. Menú principal Gesdep  
(Fuente <http://www.gesdsp.net>)

Sus funciones van desde gestionar todo el tema administrativo de un club hasta controlar el área técnica del día a día de los equipos.

Lista de Jugadores

Fichas Individuales Informes Exportar ?

Búsqueda rápida: Todos los Equipos Orden: Equipo + Nombre Depo Id Buscar Limpiar

Desde Año Nac. Hasta Año Nac. Lateralidad Cualquier Posición Solo posiciones principales Incluir equipos especiales

RELACIÓN DE JUGADORES

Con	Tit	Sup	S/J	NCo	Min	Gol	Asi	TAm	TRo	Val
29	26	3	3	2	1998	2	0	3	0	3,79

17. Alberto JUV

Lopez Díaz, Alberto

19/02/1997 (21 años)

615476761

cesar7aw7@gmail.com

Casado JUV

Casado Pérez, Antonio

17/04/1998 (20 años)

615384522

Antonio36@gmail.com

César JUV

Bolívar Puerta, César

13/02/1997 (21 años)

615404937

Alberto24@gmail.com

Fabio JUV

Bolívar Tabares, Fabio

17/01/1998 (20 años)

615433570

Fabio27@gmail.com

Figura 15. Gestión de equipos Gesdep  
(Fuente <http://www.gesdep.net>)

Nombre de la animación: Estrategia Falta 1



Figura 16. Pizarra táctica Gesdep  
(Fuente <http://www.gesdep.net>)

- Ventajas de Gestión Deportiva:
  - Incluye sección financiera para controlar ingresos y gastos de la entidad deportiva.
  - Gestión de los equipos del club, pudiendo llevar un control y registro de informes, estadísticas, etc.
  - Gestión de competición, donde registrar todos los resultados de los equipos, así como clasificaciones etc.
  - Planificación de sesiones de entrenamientos, pizarra táctica, diseñador de entrenamientos...
- Desventajas:
  - Interfaces pobres, larga curva de aprendizaje para saber dónde tienes que ir para hacer lo que quieres hacer.
  - Pobre movilidad, imposibilidad de probar la aplicación en el móvil, solo siendo accesible a ver sus funcionalidades desde la versión de escritorio.
  - Pizarras tácticas y diseñadores de ejercicios que cumplen su cometido, pero son complicados de usar y en muchas ocasiones no sabes cómo continuar o hacer cambios.
  - Pago por suscripción, de forma que en el momento dejes de pagar perderás toda la información y datos almacenados.

### 3.4. Limitaciones de los sistemas

Tras realizar este estudio de las herramientas existentes del mercado, vemos ciertos patrones en cuando a limitaciones de las funcionalidades que pasamos a analizar.

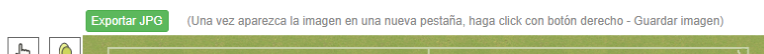
#### 3.4.1. Interfaces pobres

La tónica general de las aplicaciones del mercado son interfaces de usuario pobres que no se han adaptado a los estándares actuales. Aplicaciones como Gesdep o SportEasy proveen una funcionalidad bastante clara a priori. Lo que ocurre con frecuencia es que nos perdemos a la hora de querer hacer una tarea o encontrar un dato concreto. Como apunta Pamela Rodríguez en su artículo “La Importancia de la Claridad y Sencillez de una Interfaz de Usuario” [6] si el usuario se traba en una tarea y no logra encontrar lo que quiere, probablemente cambie de lugar o aplicación, perdiendo un consumidor potencial.

Un ejemplo lo vemos en gesdep.net, donde la tarea de añadir un ejercicio diseñado a un entrenamiento consta de los siguientes pasos:

1. Desde el menú principal, ir al apartado “diseñador de ejercicios”.
2. Desde el diseñador, crear y exportar el ejercicio a formato imagen (limitaciones técnicas incluidas como podemos ver en la figura 17).
3. Ir al apartado “planificación de sesiones”.
4. Acceder a la opción “Nueva Sesión”.
5. Guardar unos datos básicos de la sesión.
6. Añadir tareas en las cuales podemos utilizar los ejercicios creados en el paso 2.

Nota importante: En la última actualización, Google Chrome impide la apertura de la imagen en una ventana nueva (botón exportar jpg). Estamos tratando de solucionar el problema, mientras, puede utilizar Firefox, que permite la exportación. Disculpen las molestias.



*Figura 17. Limitación exportación Gesdep  
(Fuente <http://www.gesdep.net>)*

Otro ejemplo lo podemos ver con la aplicación Easy2Coach, la cual requiere una serie de pasos costosos para poder diseñar y añadir tus ejercicios:

1. Desde el menú principal, seleccionar Nuevo ejercicio.
2. En la interfaz de nuevo ejercicio, vemos varios errores de funcionalidad bastante graves (listas desplegables que no se despliegan, formularios mal organizados, etc.). En esa interfaz, tenemos dos grandes botones en azul como muestra la figura 18, la cual nos abre nuevas ventanas tanto para crear imágenes estáticas como animaciones.

3. Una vez diseñados los ejercicios (con unos diseñadores bastante completos) la única alternativa que tenemos es capturar la pantalla y almacenarlos. Ya que las opciones que tenemos “in situ” hacen caer en saco roto nuestras creaciones, no almacenándose ni añadiéndose al nuevo ejercicio de ninguna manera.

**NUEVO EJERCICIO** Ayuda

**Datos del ejercicio**

Nombre:  Idioma: Español

Estructura del entreno\*  Duración\*

Edad  Número de jugadores

Tipo de entreno  Nivel de habilidad

☒ Ejercicio público

☒ Visible para todo el equipo

Campos marcados (\*) son obligatorios

**Detalles**

Organización:

**B** **I** **U** Formats

Proceso:

**B** **I** **U** Formats

**Gráficos/Archivos** **Otros criterios**

**Haz clic para empezar Software gráfico**

**Haz clic para empezar Software animación**

Subir archivo:

☒ Confirmo que tengo los derechos de autor de esta imagen / archivo o que voy a usarlo únicamente de manera interna en mi equipo sin publicarlo para ningún otro usuario.

Figura 18. Formulario y diseñador de ejercicios Easy2Coach  
(Fuente <https://easy2coach.net>)

En definitiva, la tarea de poder diseñar tus ejercicios para almacenarlos y añadirlos a sesiones de entrenamiento (objetivo más o menos principal de todas las aplicaciones y sistemas revisados) se complica bastante para el usuario no habitual de estas herramientas. Además, en casos como este último, la solución que se nos ofrece nos lleva a la confusión ya que no sabemos si vamos a poder insertar un ejercicio directamente desde el diseñador o si es necesario realizar capturas de pantalla para poder exportar y usar tus ejercicios.

### 3.4.2. Usabilidad de los diseñadores

Otro de los puntos críticos de las aplicaciones estudiadas es la usabilidad de los diseñadores de ejercicios. Existen varios problemas que tienen todas las aplicaciones en común y que desglosaremos en detalle.

Antes de entrar en detalle con el resto, tenemos una aplicación bastante antigua tecnológicamente hablando (CampusDeportivo) que ofrece una solución y forma de hacer las cosas bastante arcaica. Hablamos de una aplicación clásica de los Windows antiguos, con diálogos de confirmación para cada acción. Por otra parte, también se le da muy poco espacio al diseñador dentro de la aplicación a pesar de ser el elemento principal de esta (figura 19).



Figura 19. Diseñador pequeño de Campus Deportivo  
(Fuente <http://www.campusdeportivo.com>)

A continuación, pasamos a desgranar los patrones de usabilidad del resto de aplicaciones deportivas cuyo diseñador de ejercicios aporta valor al producto (recordemos que SportEasy no posee diseñador de ejercicios), en la cual veremos repeticiones y problemas comunes entre todas ellas que rompen con los principios de usabilidad definidos por Jakob Nielsen [7]:

- Para empezar, ninguna de ellas es lo suficientemente flexible como para que pueda ser usada por todo tipo de usuarios. Como hemos explicado, ninguna de las aplicaciones estudiadas ofrece una forma intuitiva o fácil de realizar las cosas, por lo que requiere una larga curva de aprendizaje y un conocimiento previo del mundillo y de la plataforma. Estas características hacen que no se cumpla el principio de Flexibilidad y eficiencia de uso.
- Además, la mayoría de pantallas que nos encontramos en Gesdep, Campus Deportivo o Soccer Specific, contienen una gran cantidad de opciones e información, muchas de las cuales prácticamente ni se usan, tapando así algunas de las funcionalidades más importantes (como el dibujo de elementos sencillos). Esto también rompe con los diseños estéticos y minimalistas propios de los principios de usabilidad.
- Por poner un ejemplo más, a la hora de realizar ciertas tareas nos hemos encontrado con aparentes cuelgues de las aplicaciones (como en el diseñador de ejercicios de Easy2Coach). O simplemente podía ser que no sabíamos el estado en el que se encontraba la plataforma, lo cual genera confusión y aumenta el riesgo de rebote de los usuarios.

En resumen, nos encontramos ante unas aplicaciones desfasadas tecnológicamente hablando, ya que a pesar de que cumplen con su funcionalidad, no se han adaptado a los estándares actuales de usabilidad y accesibilidad. Por ello, estas aplicaciones pueden llegar a un target de usuarios mucho más reducido que el que podrían llegar a tener si aplicaran unas pautas y estrategias de diseño más consonantes con las tendencias actuales [8].

#### 3.4.3. Precios y modelos de negocio de los sistemas

Por último, otro de los grandes inconvenientes de las aplicaciones del mercado son los modelos de negocio que adoptan, siendo poco flexibles a la hora de abrir sus puertas a nuevos usuarios.

Nos encontramos con CampusDeportivo, un sistema de pago 100%, el cual para utilizarlo es necesario adquirirlo mediante un pago único. Sólo en ese momento podremos hacer uso de él. Este es un gran inconveniente ya que, según las tendencias actuales en cuanto a modelos de negocio en internet, este tipo de monetización está en decadencia y genera poca adopción de usuarios. En el polo opuesto encontramos las aplicaciones gratuitas, que dan acceso a todo el mundo tras un simple registro. El problema de estas aplicaciones es que son también las más limitadas en cuanto a diseño de ejercicios y gestión de entrenamientos. Dentro de este grupo tenemos a SportEasy, sin un diseñador de ejercicios entre sus funcionalidades y a Easy2Coach, con un diseñador de ejercicios bastante limitado en cuanto a comodidad de uso y utilidad. Por parte de estas aplicaciones es un punto positivo el tener la oportunidad de probarlas antes de tomar una decisión sobre invertir o no en estos servicios.

Por último, encontramos aplicaciones como GestionDeportiva o SportEasy, con modelos de negocio un tanto restrictivos. Si bien es cierto que con GestionDeportiva podemos acceder a una demo estática de la aplicación (con datos de prueba, ejercicios diseñados, etc.), no podemos realizar ninguna prueba de funcionalidad, añadir nuevos datos, crear nuevos ejercicios, etc. Para poder realizarlo, tendría que ser previo contacto con la empresa para poder crear nuestra instancia de datos y poder empezar a trabajar.

En cuanto a SportEasy, las restricciones van más allá. Es cierto que ofrecen periodos de prueba gratuitos de 7 días, pero no sin antes introducir tus datos de la tarjeta de crédito, por lo que un usuario de internet que esté buscando una herramienta de este estilo probablemente probará otras plataformas antes de introducir su tarjeta de crédito en portales desconocidos para él.

En conclusión, la tónica general de estas aplicaciones es que ninguna de ellas ayuda al usuario a tener una adopción fácil de las aplicaciones, poniendo barreras ya sea en forma de restricciones de funcionalidades o introduciendo métodos de pago de forma prematura.



### 3.5. Tecnologías para el desarrollo del proyecto

En cuanto al estado de las tecnologías actuales, vamos a realizar tres secciones distintas para analizar por separado aquellas tecnologías que resuelvan problemas distintos. Para ello, empezaremos analizando tecnologías de persistencia de datos, donde veremos dónde se encuentra el mercado y qué herramientas nos ofrece. Segundo, veremos las tecnologías de servidor o “back-end” y cuál es el estado del mercado en la actualidad. Por último y muy importante también, analizaremos los distintos frameworks de desarrollo web o “front-end” y analizaremos cual se ajusta más a las necesidades de nuestro proyecto.

#### 3.5.1. Persistencia de datos

Tradicionalmente, la persistencia de datos en sistemas multimedia o sistemas de información se ha realizado mediante bases de datos relacionales (SQL) a través de distintos gestores (SQL Server, Oracle, ect.). Sin embargo, desde mediados de los 2000 han surgido tipos de bases de datos no relacionales, las cuales aportan más flexibilidad y rapidez a ciertas estructuras de datos. Estas últimas son las que más han crecido estos últimos años, coexistiendo actualmente ambas en el mercado. Pasamos a detallar los distintos tipos de bases de datos.

##### 3.5.1.1. Bases de datos relacionales

- Este tipo de bases de datos utilizan el lenguaje de comunicación estándar y más extendido de las bases de datos: SQL. Siendo el principal cambio entre las distintas implementaciones el gestor de base de datos usado.
- Dentro de estos tipos de bases de datos se definen esquemas de datos únicos y rígidos, que requieren de una concienzuda fase de diseño previa implementación.
- Se rigen por los principios ACID [9]:
  - Atomicidad: Garantiza que cuando se realiza una transacción con múltiples pasos, se garantiza que todos ellos se ejecuten o que no se ejecute ninguno. Esto permite un buen tratamiento de los errores y evita que se corrompan datos.
  - Consistencia: También llamada integridad, que asegura que todas las transacciones que se llevan a cabo en una base de datos son válidas y cumplen las reglas de integridad de la base de datos.
  - Aislamiento (Isolation): Asegura que operaciones atómicas no afecten a otras transacciones, gracias a que la base de datos se encarga de manejar cómo y cuándo se producen los cambios de una operación en caso de concurrencia.

- Durabilidad: Lo cual asegura que una vez se realiza una operación, esta no se podrá deshacer y los datos persistirán a pesar de cualquier cambio.

#### 3.5.1.2. Bases de datos no relacionales

La principal diferencia de las bases de datos no relacionales, es que no requieren un esquema de datos previo para empezar a almacenar información, ya que pueden almacenar cualquier tipo de datos con cualquier estructura. Esto es útil cuando no sabemos de antemano la información o el formato que vamos a necesitar almacenar. Normalmente, los datos que almacenan las bases de datos no relacionales vienen en formato JSON.

Además de estas características, los datos almacenados en las bases de datos no relacionales se pueden visualizar y consumir de diferentes maneras, en función de las necesidades [11].

#### 3.5.2. Tecnologías Back-end

Las tecnologías backend (aquellas utilizadas en el lado del servidor), tienen como objetivo gestionar las peticiones de información que le llegan, tratar esas peticiones, comunicarse con bases de datos y en última instancia devolver una respuesta al dispositivo emisor de la petición.

Dentro de estas tecnologías, tenemos lenguajes clásicos con un amplio recorrido a través de los años, como PHP o Java, y otros lenguajes más modernos que han surgido recientemente, como Javascript y Python. Dentro de cada una de estas tecnologías, existen diferentes frameworks o “marcos de trabajo” que revisaremos a continuación.

##### 3.5.2.1. PHP

El lenguaje PHP es el más extendido en el mundo de la programación web. Esto se debe en parte a que era el más utilizado hace unos años y a que casi todos los CMS del mercado (incluido Wordpress) lo utilizan.

Es un lenguaje con muchos amantes y detractores, y actualmente el principal uso que se le da es a través de Laravel, el framework PHP más usado para su uso en el servidor, el cual posee una gran comunidad detrás y presenta las siguientes características [12]:

- Aplicación del lenguaje PHP con el modelo MVC (Modelo-Vista-Controlador), comúnmente usado en web.
- Dentro del modelo MVC, las vistas se crean con Blade, que es un sistema de plantillas que se pueden extender y personalizar a medida mediante HTML o CSS.
- Uso de un sistema ORM (Eloquent) para gestionar las consultas a las bases de datos, de forma que se obtiene una capa intermedia que aporta seguridad. Esto se correspondería con los modelos del MVC.

- Estas características, junto con los controladores, potencian Laravel de forma que se obtienen productos mucho más escalables y mantenibles que con otros lenguajes nativos.
- Además de sus características técnicas, Laravel posee una gran comunidad detrás por lo que ejemplos y documentación a la hora de desarrollar no se van a echar en falta, como por ejemplo a la hora de realizar integraciones con otros sistemas como Google, ERP's, etc.

#### 3.5.2.2. *Java*

Java es sin duda uno de los lenguajes más extendidos y con más recorrido dentro de la programación de aplicaciones. Esto se debe a diversas características que lo distinguen de otras alternativas [13]:

- Se trata de un lenguaje muy escalable, sus opciones de empaquetamiento y de desarrollo de componentes hace que, una vez construida la aplicación, esta sea fácilmente escalable añadiendo nuevas instancias o recursos.
- Es un lenguaje multiplataforma y puede desplegarse en la mayoría de plataformas como Windows, Mac OS, Linux, Android, etc. Por lo que a nosotros ocupa, los archivos Java pueden ejecutarse en muchos servidores distintos, sin importar el sistema operativo usado.
- La gestión de memoria es otro punto muy importante en el lenguaje Java, haciendo un uso eficiente de los recursos disponibles.
- Otra característica es el “multi-hilo”, lo cual permite que una aplicación sea ejecutada simultáneamente por diferentes usuarios. En la actualidad existen más opciones con ejecución “multi-hilo”, pero Java fue la primera en implementarlo.
- Sin embargo, todas estas características hacen que construir y ejecutar una aplicación desde 0 sea más costoso con Java que con otros lenguajes disponibles, debido a su mayor complejidad de desarrollo y de configuración de la aplicación.

#### 3.5.2.3. *Python*

El lenguaje Python tiene un largo trayecto ya que surgió en 1989, pero es en los últimos tiempos cuando más popularidad está adquiriendo, sobretodo por su cantidad y calidad de frameworks de desarrollo: Flask, Django, etc. Entre las características más importantes de este lenguaje encontramos [14]:

- Es un lenguaje más fácil de escribir y entender que sus competidores. Su sintaxis sencilla hace que leer en Python sea como leer un texto en inglés, por lo que se reducen las curvas de aprendizaje.
- Echar a andar un proyecto en Python debería ser muy sencillo, permitiendo construir ejemplos o prototipos de forma rápida.
- Tiene una gran comunidad de desarrolladores detrás, impulsada por su concepto de Código Abierto.
- Es un lenguaje interpretado, no compilado, y su velocidad a la hora de ejecutar código es muy alta.

#### 3.5.2.4. *Javascript*

Sin duda, JavaScript es el lenguaje más importante a la hora de hablar de desarrollo web. A sus amplias soluciones a la hora de elaborar interfaces de usuario y funcionalidades se le suma el auge de JavaScript para aplicaciones en la parte del servidor/back-end. Existen numerosas alternativas para desarrollar back-ends sólidos con JavaScript y todas ellas adoptan las ventajas y flexibilidad que ofrece este lenguaje. Hablamos de Node.JS, Modernizr, Express, Ember JS, Backbone JS, Koa, Meteor, Hapi, etc. Entre sus características más importantes tenemos [15]:

- Estructura más adecuada para el desarrollo web, ya que es un lenguaje orientado a eventos. Esto quiere decir que trabaja con asincronía, conforme recibe las peticiones estas se van procesando de forma asíncrona sin bloquear otras tareas. Esto provoca que JavaScript sea muy ágil a la hora de gestionar eventos.
- Además, tiene una estructura ligera con muy poco peso, por lo que hace que pueda ser instalado en cualquier instancia o servidor.
- Al ser el mismo lenguaje usado por desarrolladores front-end, se potencia la figura del desarrollador full-stack, que puede desarrollar tanto funcionalidades front-end como funcionalidades back-end, aumentando la versatilidad dentro de los equipos de desarrollo.
- En definitiva, con JavaScript se pueden construir back-ends ligeros y sin largas curvas de aprendizaje añadidas.

#### 3.5.3. *Tecnologías Front-end*

A la hora de hablar del desarrollo front-end, nos encontramos ante un panorama mucho más simplificado que el de desarrollo back-end. Sigue existiendo el pack HTML, CSS y JavaScript, pero se ha profundizado bastante hasta el punto de tener diferentes frameworks, opciones y formas de trabajar a la hora de desarrollar una aplicación web. Estos frameworks están basados en

JavaScript, y junto con otras herramientas como puede ser TypeScript, hacen que el desarrollo front-end sea ahora algo mucho más estructurado.

Pasamos a revisar los frameworks de desarrollo más importantes actualmente y cual encajaría más con nuestro proyecto.

#### 3.5.3.1. *Angular*

Angular se trata de un framework JavaScript desarrollado por Google, cuya función es facilitar la creación de aplicaciones web SPA. Algunas de las características de Angular son [16]:

- Ofrece una estructura predeterminada para organizar nuestros proyectos.
- Se apoya en TypeScript, que se trata de una especie de JavaScript tipado y con algunos cambios más.
- Al igual que la competencia y los frameworks modernos, basa su estructura en componentes web, los cuales son piezas de código que se pueden insertar en cualquier sitio de la aplicación.
- Posee una gran integración con diferentes IDEs de desarrollo, lo que hace mucho más cómodo el desarrollo con este framework.
- Podría decirse que es el framework más completo, el que más herramientas o facilidades posee dentro del propio framework.

#### 3.5.3.2. *React*

React, al contrario que Angular, se trata de una librería JavaScript, en este caso desarrollada por Facebook. Decimos que es una librería y no un framework ya que React se ocupa de las interfaces de usuario y complementa su funcionalidad con otras librerías o estructuras de código, ofreciendo flexibilidad al programador. Algunas de las características más importantes de React son [17]:

- Uso del Virtual DOM, que se trata de un DOM que React guarda en memoria, por lo que es mucho más rápido acceder a él que al DOM del navegador. Entonces, cada cambio que se produce, React lo refleja en el DOM Virtual, momento en el que compara ambos DOMs y renderiza lo que haya cambiado en el DOM del navegador. Todo esto lo hace de manera transparente para el desarrollador y produce un mayor rendimiento de la aplicación.
- Es isomórfico, por lo que el código que se genera en el navegador, también se renderiza en el servidor (al contrario que con otras librerías JavaScript o JavaScript nativo). Lo que le ocurre a JavaScript con los motores de búsqueda es que, si el contenido de una página

se genera en la parte del cliente, los indexadores leerán la página como una página vacía. Esto es lo que soluciona el isomorfismo de React.

- Por último, es una librería extensible que puede ser complementada con muchas otras librerías JavaScript, permitiendo diseñar y crear soluciones a medida de forma flexible.

#### 3.5.3.3. *VueJS*

Vue se trata del framework JavaScript que ha surgido y se ha popularizado más recientemente. Lo más interesante de él es que se puede utilizar tanto para pequeños componentes como para proyectos más complejos. Entre sus características encontramos [18]:

- Es un framework progresivo, lo que quiere decir que podemos empezar a utilizar Vue en proyectos o páginas web existentes con solo importar una librería.
- Además, permite la creación de proyectos más grandes, pudiendo extender su funcionalidad a igual que con Angular y React.
- Al igual que sus competidores de Google y Facebook, ofrece una estructura de componentes a la hora de construir interfaces, por lo que continúa siendo una opción igual de válida a la hora de construir proyectos desde cero, ya que su estructura y funcionalidades son similares.

#### 3.5.4. Conclusiones y tecnologías seleccionadas

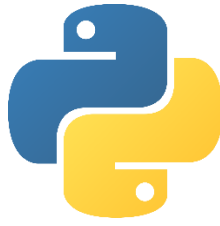
Tras analizar las cualidades de cada tecnología y atendiendo a las características del producto que se quiere construir se usarán las siguientes herramientas para el desarrollo del proyecto:

- Para la persistencia de datos se usará una base de datos relacional (SQL). Nos decantamos por esta opción ya que nuestro proyecto requiere de: autenticación de usuarios, relaciones de entrenadores con entrenamientos/ejercicios, pertenencia de ejercicios a varios entrenamientos, etc. Una base de datos no relacional con requisitos de este tipo puede añadir complejidad innecesaria a nuestro desarrollo.



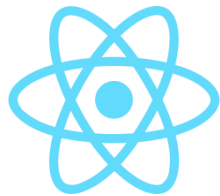
*Figura 20. Bases de datos SQL*

- Para el back-end se ha elegido Python como tecnología de desarrollo. Su popularidad creciente, su desarrollo simplificado y sus particulares características han determinado la elección, más aún al tratarse de un back-end simple en forma de API cuya función principal será conectarse a la base de datos.



*Figura 21. Lenguaje Python*

- Por último, para el front-end la tecnología elegida es React. Sin duda es la tecnología que más mercado posee respecto al mundo del desarrollo web. Además, su flexibilidad y extensibilidad hacen que sea posiblemente la mejor opción para afrontar retos como el diseñador de ejercicios, la aplicación multiplataforma, etc. [19]



*Figura 22. ReactJS*

## 4. Objetivos

Así pues, queremos plantear un objetivo que se centre en solucionar los principales problemas de las aplicaciones del mercado en cuanto al diseño de ejercicios y planificación de entrenamientos.

Para ello, se quiere obtener una aplicación mobile-friendly que cumpla ciertos criterios de usabilidad y accesibilidad tanto en dispositivos móviles como en entornos de escritorio. Para ello, y gracias a las tecnologías seleccionadas, lo que haremos será construir una base de datos SQL y un backend con Python común para nuestras versiones de móvil y de escritorio. Evidentemente, a la hora de construir nuestro frontend, sí que deberemos de tener en cuenta las diferentes versiones que estamos desarrollando, es por ello por lo que usaremos extensiones de React para poder construir una aplicación verdaderamente usable y multiplataforma, donde podremos reutilizar la lógica común y gran parte del código y así centrarnos en potenciar las particularidades de cada plataforma [20].

Así pues, tendremos una aplicación fácil de usar que nos permitirá sacar partido de sus virtudes rápidamente:

- El acceso a la aplicación no será un impedimento, se quiere maximizar el ratio de gente que se registra o inicia sesión respecto a la que visita nuestro sitio, ya sea a través de un método convencional de registro o a través de OAuth con los servicios de autenticación de las principales redes sociales.
- El diseño de ejercicios será simple y accesible, sin importar desde qué dispositivo accedamos, siempre tendremos la experiencia de usuario adecuada a las necesidades del consumidor (por ejemplo, tendremos un diseñador simplificado o reestructurado a la hora de ser usado en dispositivos móviles, ya que en lugar de clicks, estos dispositivos se manejan por gestos).
- Por último, dar visión en todo momento de que queremos ayudar a los usuarios y que sacar un beneficio de ellos no es un objetivo sino una consecuencia. Acceder a nuestra aplicación será gratuito y se podrá utilizar sin problema, sin embargo, existirán pequeñas funcionalidades que aporten más valor a la aplicación a cambio de una reducida cuota anual o mensual.

En definitiva, queremos construir una aplicación moderna, simple y que cumpla su cometido con un nivel de calidad adecuado a las necesidades concretas de estos usuarios.



## 5. Planificación y metodología

Desde un inicio, la idea principal era seguir una metodología de trabajo ágil para el desarrollo del proyecto. Sabíamos de la importancia de una buena planificación y de la metodología que se debe escoger para el desarrollo, ya que estas pueden determinar el transcurso exitoso o errático de un proyecto. Tras analizar las metodologías más populares del mercado para la planificación de proyectos, hemos visto que algunas metodologías clásicas de gestión de proyectos, como por ejemplo Waterfall, provocan diversos problemas al ofrecer soluciones rígidas y poco adaptables a los cambios (tan presentes en el desarrollo de software) [21]. También se ha visto que otras como SCRUM nos ofrecen una forma versátil y abierta de organizar nuestros proyectos, incluso para proyectos de carácter personal como el que nos ocupa [22].

Es por ello que para la planificación del proyecto se ha escogido la metodología ágil SCRUM. Dicha metodología nos permite tener una visión global y constante de la situación del proyecto, además de facilitarnos la identificación de problemas y retrasos de forma bastante efectiva.

El seguimiento global del proyecto lo realizaremos con la ayuda de [Trello](https://trello.com), una herramienta online para administrar proyectos basada en la organización de tableros, tarjetas y listas de tarjetas. Con Trello, tendremos visión en todo momento del estado del proyecto, tareas futuras, tareas pendientes, tareas en curso y tareas finalizadas.

Para gestionar nuestro proyecto, siguiendo la metodología SCRUM, se realizarán planificaciones cada dos semanas, en las que se escogerán tareas que aporten más valor o funcionalidad a la herramienta y se estimarán las horas que puede llevar realizar dichas tareas. Estas planificaciones dan como resultado una “iteración” o “sprint” como el de la figura 23, de dos semanas de duración, en las que se añadirán tareas hasta llegar a las 20 horas de estimación. ¿Por qué 20 horas? 20 horas es la cantidad de tiempo escogida para dedicar al proyecto en un periodo de tiempo de dos semanas.

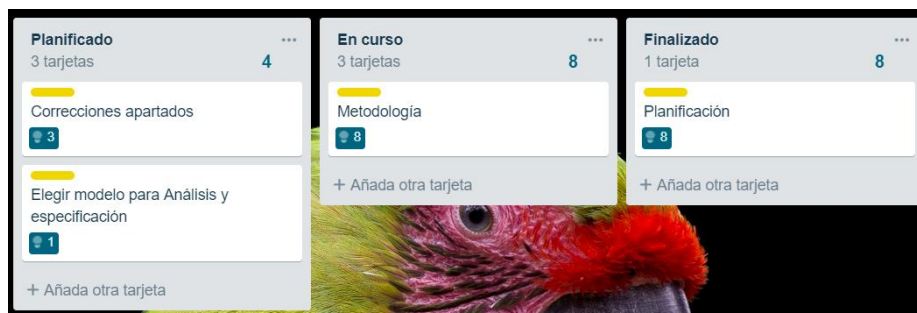


Figura 23. Sprint de trabajo actual  
(Fuente <https://trello.com>)

Durante la iteración, el flujo de trabajo será el siguiente:

- Se irán cogiendo en orden las tareas planificadas para la iteración, con forme se cojan se pasan a estado en curso. El punto ideal es que no se pasen a en curso tareas mientras existan otras en ese estado.
- Una vez finalizadas las tareas, estas se pasarán de en curso al estado finalizadas.

Con estos tres estados (Planificadas, En curso y Finalizadas) tendremos visión en todo momento del trabajo realizado hasta el momento y el % de horas completado, por lo que podemos ver si llevamos retraso (por ejemplo, si al empezar la segunda semana del sprint nos queda un 70% de horas por consumir querrá decir que llevamos cierto retraso en la iteración).

Una vez finalizadas las dos semanas, obtenemos un feedback automático muy valioso. Ahí podemos comprobar si nuestras estimaciones realizadas dos semanas atrás han sido correctas, y de esta forma refinar nuestra forma de estimar las tareas en caso de que existan sobreestimaciones o subestimaciones. Por otra parte, podemos identificar bloqueos o problemas en el proyecto: en caso de que existan tareas que no se hayan podido finalizar durante la iteración, se podrán tomar las medidas pertinentes para evitar que se enquiste la situación (volver a analizar la tarea y reestimarla, buscar algún tipo de ayuda externa, etc.). Por último, podemos comprobar si nuestro ritmo de trabajo ha sido el adecuado o si es necesario algún cambio a la hora de invertir tiempo en el proyecto.

Una vez realizadas las primeras tareas y planificaciones iniciales, en el momento de la redacción de este apartado se han llevado a cabo el 10% de las tareas, invirtiendo en ellas 30 horas de las 300 del proyecto. Eso quiere decir que quedan 270 horas restantes a partir del momento actual para la finalización del proyecto. Estas 270 horas se dividirán en 14 iteraciones de 20 horas, empezando la primera de ellas el 19 de noviembre y terminando el 2 de diciembre y acabando con una última iteración del 20 de mayo al 2 de junio, teniendo un margen de 10 horas ante posibles desviaciones o problemas.

La propia metodología SCRUM dicta que, a pesar de la visión global del proyecto, hasta la planificación de cada iteración no se deciden las tareas que se van a realizar en el sprint. En cada planificación de iteración, se procederá al desglosado (en caso de que sea necesario), estimación y planificación de las tareas, determinando los próximos 14 días de trabajo.

En nuestro caso, tenemos un “backlog” sin desglosar, o lo que es lo mismo, un listado de tareas definidas a muy alto nivel que contienen una gran cantidad de horas de trabajo. El objetivo de cada planificación será ir desglosándolas y añadiéndolas al sprint actual.

## 6. Análisis y especificación

### 6.1. Introducción

Para el análisis y especificación de requerimientos de nuestro proyecto se hará uso del estándar IEEE 830 para la Especificación de Requerimientos de Software (ERS).

#### 6.1.1. Propósito

El presente apartado tiene como objetivo registrar, definir y documentar los requisitos de nuestra aplicación, es decir, aquellas funcionalidades que deberá ofrecer finalmente la aplicación.

#### 6.1.2. Ámbito del sistema

Nuestra aplicación se trata de un proyecto que pretende facilitar la vida de los entrenadores de fútbol no profesionales facilitando sus gestiones y trabajos diarios.

Para ello, el servicio se compondrá de las siguientes partes:

- Diseño de ejercicios, con interfaces sencillas y con sets de elementos predefinidos para poder hacer los ejercicios de una forma clara. Con esto se quiere conseguir que los ejercicios estén siempre disponibles para los entrenadores para su reutilización, clasificación por tipos, etc.
- Diseño de entrenamientos, de forma que se pueda planificar un entrenamiento de forma fácil, incluyendo opciones del estilo de cantidad de jugadores disponibles para el entreno, los ejercicios que se van a realizar, su duración, etc.
- Plantillas de entrenos, porque a lo largo del tiempo se repiten con frecuencia los mismos tipos de entrenamiento, con las mismas fases o con ejercicios en común. Es por ello que realizar siempre un entrenamiento desde 0 al final resulta en mucho tiempo invertido repitiendo acciones una y otra vez. Para este problema, las plantillas de entrenamientos son una buena herramienta que los entrenadores pueden a tener a su disposición y así verse facilitado su trabajo.
- Históricos de los entrenamientos y del trabajo realizado. De forma que los entrenadores puedan tener siempre accesible el trabajo realizado a lo largo de los meses y de las temporadas (para ver evoluciones, analizar fallos, etc.)

### 6.1.3. Definiciones, Acrónimos y Abreviaturas

- API: Interfaz de Programación de Aplicaciones.
- IEEE: Institute of Electrical and Electronics Engineers.
- Front-end: Parte de las aplicaciones web que se ejecuta en la parte del cliente (los navegadores).
- Back-end: Parte de las aplicaciones web que se ejecutan en la parte del servidor y no son visibles directamente para los usuarios.
- Framework: Marco de trabajo, normalmente dentro de alguna tecnología o lenguaje de programación concreto.
- HTML: Lenguaje de marcado usado para definir el contenido de las páginas web (Hypertext Markup Language).
- CSS: Lenguaje de hojas de estilo en cascada utilizado para definir el diseño y visualización de las páginas web (Cascading Style Sheets).
- JWT: Json Web Token, sistema utilizado para realizar autenticación en las aplicaciones y mantener la sesión a través de un web token.
- CSRF: Ataque web que fuerza al navegador web de la víctima (validado en algún servicio web) a enviar una petición a una página más vulnerable, a fin de modificarla o obtener datos sensibles [23].

## 6.2. Descripción general

### 6.2.1. Perspectiva del Producto

La idea de nuestro proyecto es que los entrenadores de fútbol tengan una especie de repositorio donde tengan almacenado todo su trabajo. Este trabajo son los ejercicios propios diseñados por los entrenadores, los entrenamientos creados, las planificaciones de trabajo a lo largo de las semanas y los meses, etc. De forma que cada vez que quieran planificar nuevo trabajo o nuevas sesiones tengan a mano todo lo necesario para realizarlo de forma cómoda.

### 6.2.2. Características de los usuarios

Usuario	Administrador
Formación	Conocimientos de la plataforma y la gestión y administración de esta
Uso	Uso general de la herramienta, además de gestionar los usuarios de la plataforma y sus incidencias con esta

Usuario	Usuario estándar
Formación	Habilidades en el manejo de ordenadores y dispositivos móviles, conocimientos de entrenamiento de fútbol, metodologías, etc.
Uso	Gestionar su cuenta, crear sus propios ejercicios, planificar entrenos o revisar el historial de trabajo realizado.

### 6.2.3. Funciones del Producto

	Administrador	Usuario estándar
Registrarse		x
Iniciar sesión	x	x
Dar de alta/eliminar usuarios	x	
Modificar usuarios	x	x
Diseñar ejercicios	x	x
Gestionar ejercicios (crear, editar, borrar)	x	x
Gestionar entrenamientos (crear, editar, borrar)	x	x
Crear plantillas de entrenos	x	
Utilizar plantillas de entrenos	x	x
Ver histórico de entrenamientos	x	x

### 6.2.4. Restricciones

Para el despliegue y uso de la herramienta deben darse una serie de requisitos:

- Necesario ordenador o dispositivo con conexión a internet.
- El servidor que aloja la aplicación debe ser capaz de atender consultas concurrentes.
- El sistema necesitará de una Base de Datos para la persistencia de datos.
- La aplicación se desarrollará con el framework de JavaScript ReactJS.
- La comunicación entre la aplicación y la base de datos se realizará a través de una API programada con el framework Django del lenguaje Python.
- El usuario final debe tener conocimiento sobre el uso básico de un ordenador y de las aplicaciones modernas de internet.
- El sistema no puede funcionar completamente de forma autónoma, ya que las 3 partes que lo componen (Front-end, API y Base de Datos) se necesitan entre ellos para proveer un servicio completo.
- La aplicación se limitará a gestionar los usuarios de la plataforma y a permitir la gestión e historial de los ejercicios y entrenamientos creados por los usuarios/entrenadores.

#### 6.2.5. Suposiciones y dependencias

Los usuarios deberán tener conexión a Internet a través de navegadores web modernos (Chrome, Firefox, Edge, etc.), dispositivos móviles con navegador o dispositivos móviles Android donde desplegar la aplicación.

Para el correcto funcionamiento de la aplicación se requerirá el uso de dispositivos con 7 años de antigüedad o más recientes.

#### 6.2.6. Requisitos futuros

El sistema en el futuro podrá integrarse con otro software de gestión de clubes deportivos, donde poder alojar allí también los entrenos y ejercicios generados en la aplicación.

### 6.3. Requisitos Específicos

#### 6.3.1. Interfaces Externas

A continuación, listaremos las interfaces de usuario y las interfaces de la aplicación con las diferentes capas de la que se compone.

##### 6.3.1.1. Interfaces de usuario

La interfaz gráfica será intuitiva con tendencia a contener pocas opciones o posibles caminos, de manera que el usuario se sienta cómodo con la herramienta rápidamente. El sistema se compondrá de las siguientes pantallas para los usuarios:

- Landing page.
- Formulario de iniciar sesión/registrarse.
- Menú principal de la aplicación.
- Perfil del usuario.
- Edición de datos del usuario.
- Visualización de entrenamientos.
- Visualización de ejercicios.
- Creación/edición de entrenamientos.
- Creación/edición de ejercicios.
- Vista global de entrenamientos planificados.
- Vista global de ejercicios creados.

En el caso de los administradores, tendrán una serie de interfaces adicionales, de cara a manejar la herramienta y realizar la gestión básica de la plataforma:

- Vista de gestión de usuarios.
- Vista de datos globales sobre el uso de la herramienta.

#### 6.3.1.2. Interfaces de comunicación

Nuestro sistema se compone de tres partes diferenciadas, front-end, back-end y base de datos. La buena comunicación entre cada una de las partes será imprescindible para el buen funcionamiento de la herramienta.

Es por ello que la aplicación front-end (ya sea web o app) se comunicará con el back-end a través de peticiones sencillas (solicitud de guardado de datos, recuperado de información, etc.). Este back-end contendrá toda la lógica de negocio y se encargará de comunicarse con la base de datos a través del gestor de base de datos para persistir y recuperar la información necesaria en cada momento.

#### 6.3.2. Funciones

Se definen los siguientes requisitos funcionales de usuario:

Número de requisito	RF1		
Nombre de requisito	Registro		
Usuario	<input checked="" type="checkbox"/> Cliente	<input type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite a cada usuario visitante darse de alta en la plataforma a través de un formulario con la información básica para el registro y así permitir empezar a utilizar la herramienta.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Número de requisito	RF2		
Nombre de requisito	Inicio de sesión		
Usuario	<input checked="" type="checkbox"/> Cliente	<input checked="" type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite a cualquier usuario dado de alta en la plataforma iniciar sesión y empezar a usar la aplicación, sea cual sea su rol.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Número de requisito	RF3		
Nombre de requisito	Gestionar usuarios		
Usuario	<input type="checkbox"/> Cliente	<input checked="" type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite al administrador dar de alta nuevos usuarios, modificarlos y eliminarlos de la plataforma.		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Número de requisito	RF4		
Nombre de requisito	Diseño sobre un lienzo		
Usuario	<input checked="" type="checkbox"/> Cliente	<input checked="" type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite a los usuarios diseñar ejercicios combinando distintas utilidades: dibujado de líneas, arrastrado de objetos al lienzo, coloreado de zonas, etc.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Número de requisito	RF5		
Nombre de requisito	Gestionar ejercicios		
Usuario	<input checked="" type="checkbox"/> Cliente	<input checked="" type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite a cada usuario gestionar los ejercicios en el sistema. Esto es, permite crearlos, editarlos y borrarlos siempre que seas el usuario poseedor. El administrador puede gestionar también ejercicios del resto de usuarios.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Número de requisito	RF6		
Nombre de requisito	Gestionar entrenamientos		
Usuario	<input checked="" type="checkbox"/> Cliente	<input checked="" type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite a cada usuario gestionar sus propios ejercicios. Esto es, crearlos, editarlos y borrarlos. Los usuarios administradores también pueden gestionar entrenamientos de otros usuarios.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Número de requisito	RF7		
Nombre de requisito	Añadir ejercicios a entrenamientos		
Usuario	<input checked="" type="checkbox"/> Cliente	<input checked="" type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite a los usuarios añadir, ordenar y especificar atributos de ejercicios dentro de los entrenamientos, de forma que se pueda componer un entrenamiento con varios ejercicios.		
Prioridad del requisito	<input checked="" type="checkbox"/> Alta/Esencial	<input type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Número de requisito	RF8		
Nombre de requisito	Guardar plantillas de entrenamientos		
Usuario	<input checked="" type="checkbox"/> Cliente	<input checked="" type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite a cada usuario, cuando está creando o editando un entrenamiento, guardar dicho entreno como una plantilla para utilizar en próximos entrenamientos.		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional



Número de requisito	RF9		
Nombre de requisito	Gestionar plantillas de entrenamientos		
Usuario	<input checked="" type="checkbox"/> Cliente	<input checked="" type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite a cada usuario, cuando está creando o editando un entrenamiento, guardar dicho entreno como una plantilla para utilizar en próximos entrenamientos.		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Número de requisito	RF10		
Nombre de requisito	Utilizar plantillas de entrenamientos		
Usuario	<input checked="" type="checkbox"/> Cliente	<input checked="" type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite a cada usuario, a la hora de crear entrenamientos, seleccionar plantillas existentes para cogerlas como base para empezar a crear el entrenamiento.		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Número de requisito	RF11		
Nombre de requisito	Buscar ejercicios		
Usuario	<input checked="" type="checkbox"/> Cliente	<input checked="" type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite a cada usuario buscar ejercicios en una especie de repositorio, donde se podrá buscar por tipo, objetivo y otros parámetros.		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

Número de requisito	RF12		
Nombre de requisito	Ver histórico de entrenamientos		
Usuario	<input checked="" type="checkbox"/> Cliente	<input checked="" type="checkbox"/> Administrador	
Descripción del requisito	El sistema permite a cada usuario revisar el historial de sus entrenamientos a lo largo del tiempo con una serie de estadísticas. Por ejemplo, tipo de ejercicio más repetido.		
Prioridad del requisito	<input type="checkbox"/> Alta/Esencial	<input checked="" type="checkbox"/> Media/Deseado	<input type="checkbox"/> Baja/Opcional

### 6.3.3. Requisitos de Rendimiento

- El sistema tendrá una alta disponibilidad, al menos el 99% del tiempo la aplicación debe de estar en correcto funcionamiento, reservando ese 1% a problemas de servidores o mantenimientos de la plataforma.
- El acceso a la aplicación debe ser fluido y correcto desde gran variedad de dispositivos (ordenadores, tabletas o móviles).
- El sistema podrá ser accedido de manera concurrente por cientos o miles de dispositivos.

#### 6.3.4. Atributos del Sistema

- Escalabilidad: La solución de implementación propuesta debe ser ampliable y relativamente sencillo de adaptar a cambios en los requerimientos o ampliación de las funcionalidades.
- Versatilidad: Los diferentes componentes del sistema deben tener un bajo acoplamiento, de forma que sean sencillamente reemplazables o reutilizables.
- Fiabilidad: El sistema debe ser robusto, evitando al máximo inconsistencias en la interfaz de usuario y manteniendo a este siempre informado del estado del sistema, teniendo una sólida gestión de errores.
- Portabilidad: El sistema será portable y utilizable por distintos dispositivos, ya sea un ordenador personal o lejos de casa con un dispositivo móvil.

#### 6.3.5. Otros requisitos

- El servidor funcionará con Apache, el cual procesará las peticiones que lleguen a través de la web para redirigirlas a nuestro componente Python.
- La base de datos SQL tendrá definidos unos usuarios con permisos concretos para evitar posibles ataques.
- El back-end saneará y filtrará todas las peticiones que le lleguen, a fin de evitar ataques como inyecciones SQL.
- El sistema tendrá autenticación por JWT. El token de cada sesión lo procesará el back-end Python, determinando si tiene validez o no. De esta forma se podrán evitar los ataques CSRF.
- El front-end React realizará validaciones en todos los inputs para que el número de peticiones erróneas que le lleguen al servidor sea mínimo.

## 7. Diseño

En este apartado del proyecto queda definida nuestra aplicación. Se definirá la persistencia de los datos, la arquitectura de la solución, sus distintas capas, la especificación de la API, las interfaces de usuarios, el estilo seguido y las pruebas o testeos que debe superar la aplicación una vez desarrollada.

### 7.1. Diseño de la persistencia

Como hemos comentado anteriormente, la persistencia de datos se desarrolla con una base de datos SQL con la siguiente estructura:

- MySQL como sistema de gestión de base de datos, ya que ofrece una serie de ventajas como su velocidad a la hora de realizar operaciones, su bajo coste para ejecutar operaciones, su alta fiabilidad a la hora de no corromper los datos, etc. [24]
- Dentro de MySQL, utilizaremos el motor de bases de datos InnoDB, el cual nos da un mejor manejo de la concurrencia que otras opciones, lo cual es muy importante en un servicio web como el nuestro [25].
- Además, la base de datos permitirá el almacenamiento de logs en caso de que se detecte un error en cualquier nivel de la aplicación, de forma que tendremos una traza con información de los errores que puedan surgir en cualquier ejecución.
- El diseño de la base de datos tiene una tabla principal de usuarios. El resto de tablas tendrán claves ajenas a la tabla usuarios, de forma que los registros tengan siempre un propietario (usuario).

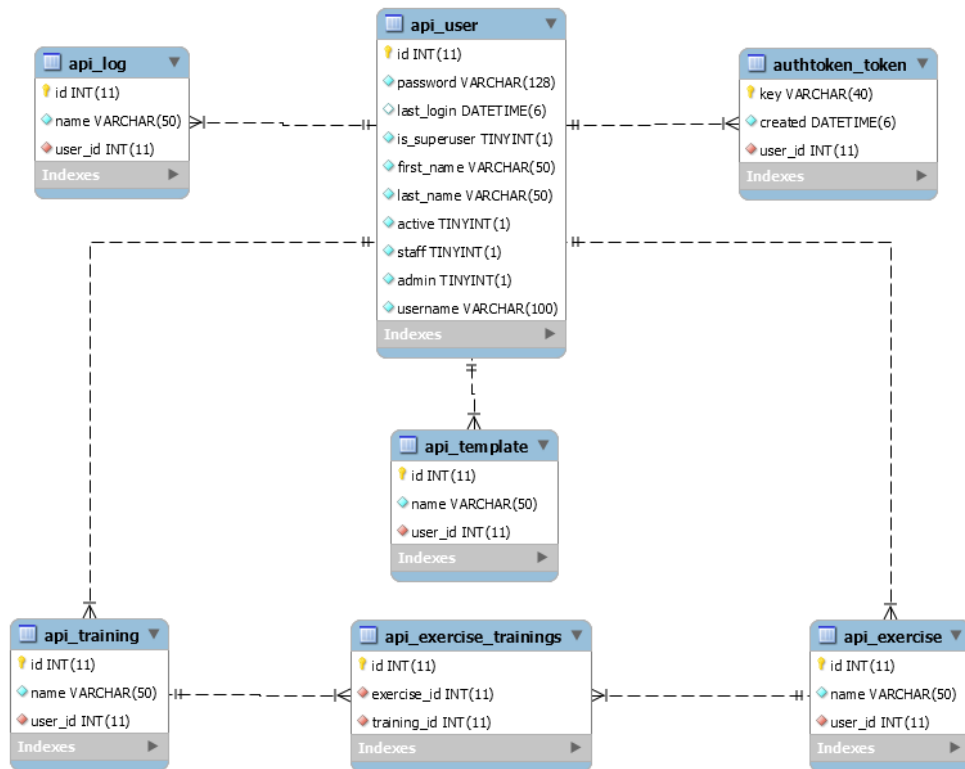


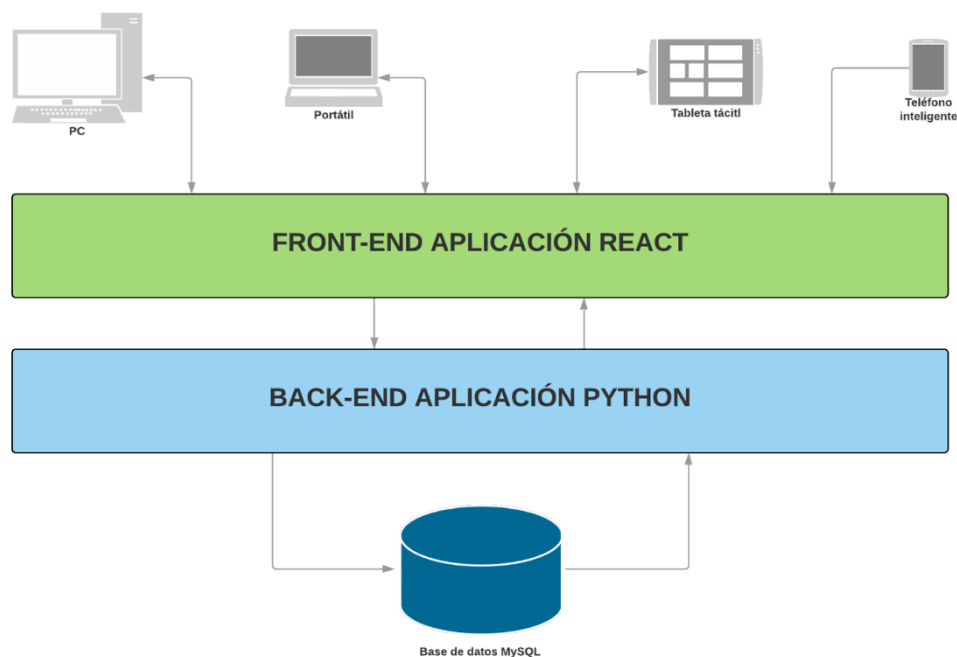
Figura 24. Modelo de datos de la aplicación  
(Fuente propia)

## 7.2. Diseño arquitectura conceptual

La arquitectura de nuestra aplicación se basa en tres componentes diferenciados: front-end, back-end y base de datos. Los usuarios, a través de sus dispositivos (pc, portátil, tableta o móvil), accederán a una URL donde interactuarán con la aplicación del front-end desarrollada con React. Para interactuar con el usuario, la aplicación front se nutrirá de datos e información que le proporciona el back-end de Python, a la vez que el front enviará los datos que necesite el back para procesar las peticiones y almacenar los datos correctamente. Por tanto, existirá una comunicación bidireccional entre ambos componentes.

Un comportamiento similar tendrá la comunicación entre la aplicación Python y la base de datos MySQL. Entre ambas, existirá una comunicación bidireccional en la que el back-end realizará peticiones de datos a la base de datos y esta devolverá los datos requeridos o el estado de la operación.

Estos comportamientos los podemos ver reflejados en el siguiente diagrama de la figura 25.



*Figura 25. Diseño arquitectura conceptual de la aplicación.  
(Fuente propia)*

### 7.3. Diseño API Rest

Nuestro back-end Python provee una API Rest que utiliza la tecnología JWT, en la que la mayoría de rutas y llamadas necesitarán un token de autenticación. Sin embargo, la API Rest también tendrá algunas llamadas en las que no será necesario el token, como por ejemplo la de iniciar sesión o la del registro. Para más información sobre el diseño de la API, ver el Apéndice I: Especificación de la API.

### 7.4. Diseño arquitectura tecnológica Front/Back-end

Como hemos comentado en el Conclusiones y tecnologías seleccionadas, las tecnologías base que definen el stack tecnológico de nuestro proyecto son SQL, Python y React. Dentro de cada una de ellas se definen los siguientes marcos de trabajo y formas de comunicación:

- **Base de datos SQL.** Nuestra base de datos usa el clásico gestor de base de datos MySQL. La particularidad aquí es que el modelo de la base de datos, así como la generación de sus tablas y campos, no se realiza a través de ningún gestor ni por línea de comandos, sino que es nuestra aplicación Python, a través de su ORM, la encargada de generar los modelos de datos y transformar estos modelos a tablas y campos SQL.

- **API Python.** Nuestra aplicación Python provee una API para que sea consumida por el front-end a fin de desacoplar la aplicación y embeber la lógica de negocio dentro de dicha API. La aplicación Python hace uso de los siguientes frameworks:
  - Django como base para el desarrollo.
  - Django Rest Framework para ayudarnos a exponer una API de comunicación.
  - MySQLClient, que sirve como interfaz para comunicar Python con la base de datos.
  - Uso del ORM de Django, para convertir las clases definidas en nuestra aplicación Django en tablas relacionales SQL.
  - Swagger para generar la documentación necesaria de la API.
- **Aplicación React.** La aplicación de React es la parte visible de nuestra aplicación, por lo que también debe tener un buen cuidado y acabado. Para ayudarnos en el desarrollo y facilitar el testeo y el despliegue en aplicaciones móviles, se han usado los siguientes frameworks y herramientas de trabajo:
  - React Native, el famoso framework de Facebook para el desarrollo de aplicaciones móvil. De esta forma, desarrollamos centrándonos directamente en la parte móvil de la aplicación. Además, como los proyectos de React Native también se pueden ejecutar y compilar en la web, estamos reduciendo duplicidades de código y aumentando la eficiencia de desarrollo.
  - Expo CLI, que nos ayuda testear y depurar nuestra aplicación en el móvil, de forma sencilla y sin tener que desplegar la aplicación ni generar paquetes .apk.

Toda esta arquitectura tecnológica la vemos reflejada en el stack tecnológico de la figura 26.

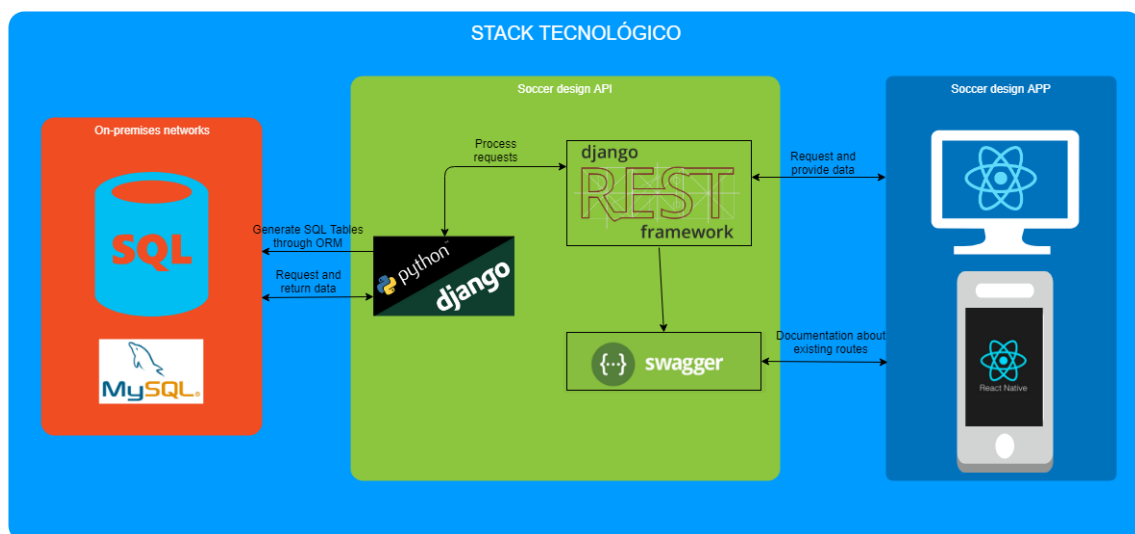


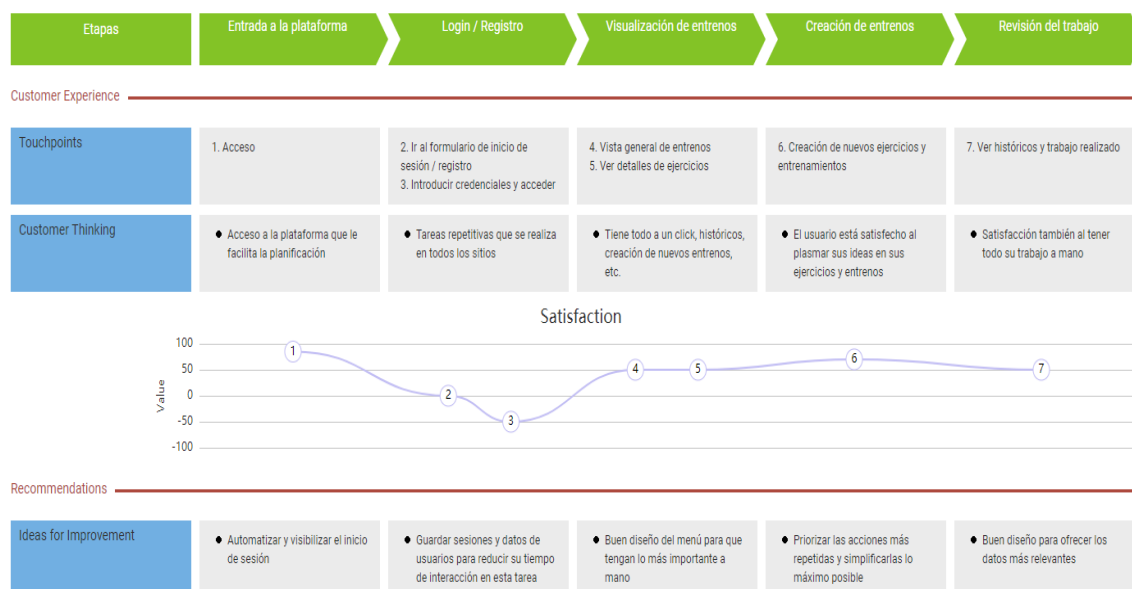
Figura 26. Stack tecnológico de la aplicación  
(Fuente propia)

## 7.5. Diseño Interacción o Experiencia de Usuario

Antes de comenzar con el diseño de las interfaces, es importante definir la interacción que van a tener los usuarios en nuestra plataforma. Para ello, nos apoyamos en la creación de un User Journey Map, en el que se definen los siguientes temas:

- La secuencia de acciones que realiza un usuario cuando entra a nuestra aplicación.
- La experiencia del cliente. Es decir, para cada acción que realiza, qué experiencia tiene el usuario (positiva, negativa, etc.).
- En qué etapas consigue el usuario sus objetivos.

Para nuestra aplicación, se define el User Journey Map en la figura 27.



*Figura 27. User Journey Map de la aplicación (Ruta habitual de los usuarios).  
(Fuente propia)*

En él, queda definido el path habitual de nuestros usuarios dentro de la aplicación, los cuales pasarán por las siguientes etapas:

1. Entrada a la plataforma, la cual será el punto de contacto inicial de los usuarios con la aplicación. Los usuarios serán recibidos con una landing page donde podrán acceder fácilmente al registro o al inicio de sesión.
2. El registro o inicio de sesión, la cual es la parte más repetitiva de todas, ya que es un sistema común que posee casi la totalidad de aplicaciones o servicios existentes en Internet. En esta parte, el usuario puede percibir la peor experiencia de todo el journey, ya que estos logins

empeoran la experiencia de usuario (de ahí la importancia de mantener sesiones de los usuarios abiertas y tener otras políticas de acceso amigables).

3. Al acceder a la aplicación en la tercera etapa, el usuario empezará a ver todo lo que le ofrece la aplicación: gestión temporal de entrenos, revisión de los últimos entrenos planificados, ejercicios creados, etc. Como tiene todo a golpe de click, un buen diseño de esta interfaz de bienvenida puede aumentar los niveles de satisfacción de los usuarios, sintiéndose agusto en esta interfaz.
4. La cuarta etapa de creación de entrenos podemos decir que es la etapa principal del journey y por la que los usuarios se registran y acceden a nuestra aplicación. Aquí crearán entrenos y ejercicios, planificarán su trabajo y podrán aprovechar su trabajo anterior para facilitar sus tareas diarias. Es por ello que el detalle y características de estas funcionalidades deben estar cuidados.
5. Por último, el usuario podrá revisar todas sus planificaciones, ejercicios trabajados y metodologías empleadas gracias al histórico de entrenamientos accesible en la aplicación. De esta manera, el entrenador puede detectar carencias en los ámbitos trabajados y establecer puntos de mejora en su metodología.

## 7.6. Diseño Interfaces

Una vez definidos los requerimientos y estudiado el journey habitual de nuestros usuarios, se procede a diseñar las interfaces de usuario de la aplicación.

### 7.6.1. Página de aterrizaje

La página de aterrizaje (figura 28) será la primera página que visualicen los usuarios. Es una interfaz pública en la que se muestra una pequeña descripción de la plataforma junto con un formulario de inicio de sesión. Somos conscientes de que los usuarios habituales el 95% de las veces entrarán a esta página para iniciar sesión, por lo que es innecesario agregar trabas al proceso. Además, a través del link “¿No eres miembro aún?” Se puede realizar un cambio de formulario para cambiar el de inicio de sesión por el de registro.



Por lo tanto, los requerimientos funcionales RF1 y RF2 de inicio de sesión y registro en la plataforma son contemplados en esta interfaz.



*Figura 28. Página de aterrizaje + apartado de login/registro.  
En esta interfaz se abarcan los RF1 y RF2.*

#### 7.6.2. Menú principal

Una vez se inicia sesión, el usuario accede al menú principal de la aplicación, la cual se divide en cuatro secciones:

- Una barra de menú común a todas las interfaces privadas, donde podemos acceder a interfaces más específicas como la lista de entrenos, la lista de ejercicios, las plantillas de entrenos, mi perfil o el menú principal.
- Un primer recuadro alineado arriba a la izquierda, que mostrará el próximo entrenamiento a realizar por el entrenador. De esta forma, el usuario ve enseguida el trabajo que tiene preparado y puede entrar a verlo en detalle con facilidad, así como editarlo, borrarlo, etc. Gracias a este recuadro, cumplimos el RF6 de gestión de entrenamientos.
- Justo debajo tenemos un recuadro que contiene un slider donde se muestran los últimos ejercicios subidos a la plataforma. De esta forma, si lo desea, el entrenador puede entrar a verlos o incluso añadirlos a sus entrenamientos. Con este recuadro cumplimos el RF11 de buscar y visualizar ejercicios.
- Por último, tenemos un cuadro a la derecha con una lista de entrenamientos (pasados y futuros). Tenemos un entreno resaltado aquí, que coincide con el que estamos mostrando en el recuadro de arriba a la izquierda. Además, pinchando en otros entrenos de esta lista, actualizaremos el mencionado recuadro, pudiendo explorar algunos detalles de estos entrenos sin salir de la interfaz. Por último, al inicio de la lista tenemos

un botón por si queremos programar un nuevo entrenamiento. Gracias a estas funciones, cumplimos el RF12 de visualizar el histórico de entrenamientos.

En resumen, desde esta interfaz podemos acceder a casi todas las tareas más repetitivas de la aplicación (visualización y creación de entrenos) de forma sencilla y efectiva, cumpliendo los criterios de usabilidad definidos anteriormente.



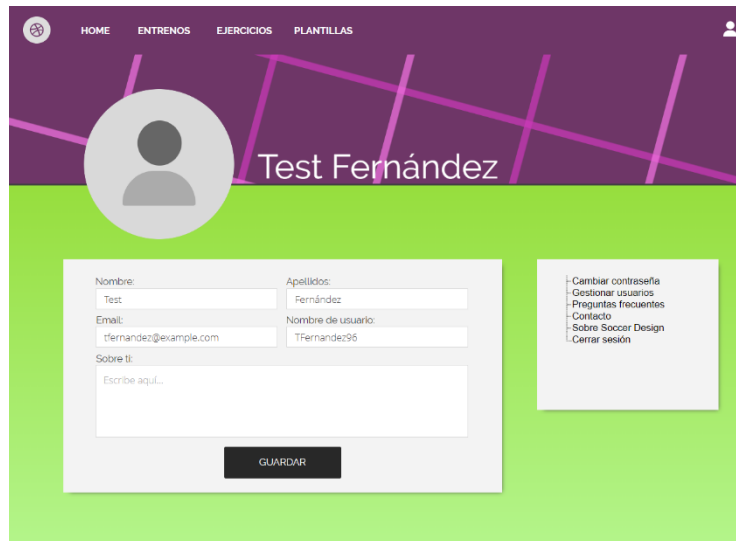
*Figura 29. Menú principal de la aplicación.  
En esta interfaz se abarcan los RF6, RF11 y RF12.*

### 7.6.3. Perfil

La interfaz de los detalles se trata de la más clásica y común dentro del mundo de las aplicaciones web/móvil. En este caso, posee una cabecera en la que destaca la foto de perfil y el nombre del usuario, con un formulario debajo donde poder ver y editar la información de nuestra cuenta.

Además, a la derecha existe un recuadro con links y funcionalidades útiles como el cambio de contraseña, la gestión de usuarios (solo para administradores), información adicional o cerrar sesión.

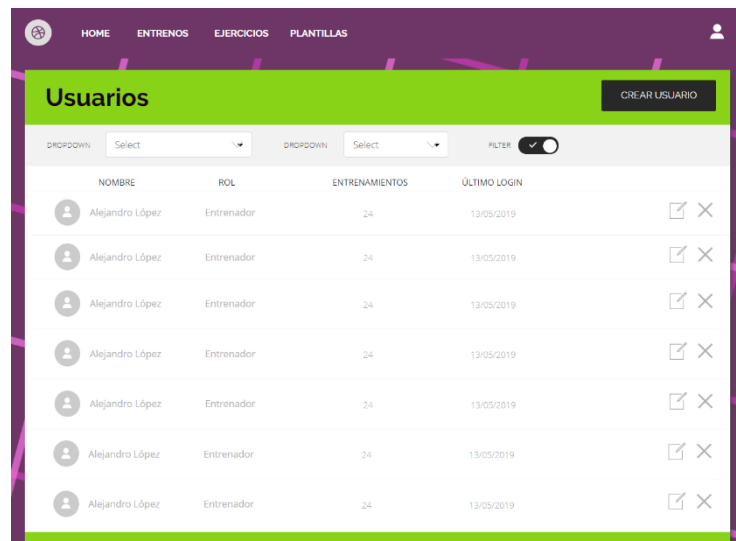
En esta interfaz se cumple parte del RF3, ya que es donde gestionas tu propio usuario.



*Figura 30. Perfil de usuario.  
En esta interfaz se abarca parte del RF3.*

#### 7.6.4. Gestión de usuarios

Interfaz clásica de administración de usuarios, donde los usuarios administradores pueden crear, editar y eliminar otros usuarios. También se pueden buscar diversos usuarios en base a parámetros como el nombre o el rol. En esta interfaz se cumple al completo el RF3 de gestión de usuarios.



*Figura 31. Gestión de usuarios.  
En esta interfaz se abarca el RF3.*

#### 7.6.5. Diseñador de ejercicios

El diseñador de ejercicios es una de las interfaces clave de la aplicación, quizá es una de las que más información y posibilidades tiene:

- Arriba del todo tenemos el título del ejercicio, el cual podemos modificar cuando estemos creando o editando un ejercicio.
- La parte principal de la interfaz es el diseñador, ubicado en el centro de la pantalla.
- A la izquierda del diseñador, tenemos un menú, el cual controlará los elementos disponibles en el menú de bajo del diseñador, de forma que no tengamos interminables listas de elementos.
- El funcionamiento básico será el arrastrado de elementos de la barra de bajo al diseñador y la colocación de estos dentro del canvas.
- A la derecha tenemos un bloque información adicional importante del ejercicio, como por ejemplo el objetivo, el número de participantes, la intensidad o la categoría del ejercicio, de forma que a la hora de buscar ejercicios en otras interfaces podamos filtrarlos de forma satisfactoria.

En esta interfaz se cumplen los RF4 y RF5 de diseño sobre lienzo y gestión de ejercicios.

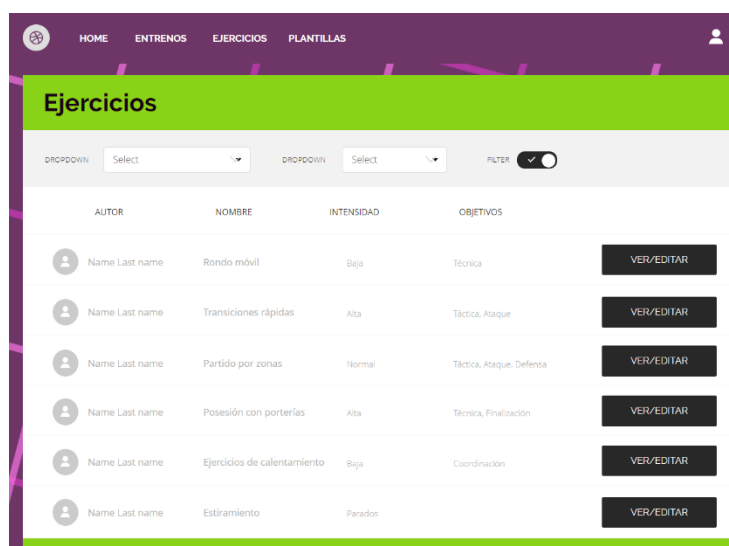


*Figura 32. Visualización/Gestión de ejercicios.  
En esta interfaz se abarcan los RF4 y RF5.*

#### 7.6.6. Lista de ejercicios

La lista de ejercicios se trata de un buscador de ejercicios en base a diferentes parámetros como categoría o intensidad. Dentro del buscador podemos entrar a ver ejercicios de otros entrenadores o entrar a modificar los nuestros propios.

En esta interfaz se cumplen los RF5 y RF11 de gestión y búsqueda de ejercicios.



*Figura 33. Lista y gestión de ejercicios.  
En esta interfaz se abarcan los RF5 y RF11.*

#### 7.6.7. Creador de entrenos

El creador de entrenos es otra de las interfaces interesantes de la aplicación. En ella podemos añadir ejercicios, determinar sus tiempos y ordenarlos, además de añadir otros detalles para tener toda la información necesaria a la hora de ejecutar los entrenamientos.

Tenemos un primer bloque a la parte izquierda donde tenemos los ejercicios a realizar en el entreno, donde podemos ordenarlos arrastrándolos a la posición adecuada, expandirlos para ver sus detalles, eliminarlos o añadir nuevos.

El bloque de la derecha contiene la información adicional del ejercicio como su descripción, objetivo principal o tipo de entrenamiento.

Además, arriba a la derecha tenemos el botón de usar plantilla, donde podemos acceder mediante un popup a una lista de plantillas donde elegir cualquiera. Después de seleccionarla, se nos cargará en nuestro entrenamiento y podemos personalizarla a nuestro gusto o necesidad.

En resumen, esta interfaz cumple con los RF6, RF7, RF8 y RF10 de añadir ejercicios a entrenamientos, utilizar plantillas en entrenamientos, guardar plantillas y gestionar entrenamientos.

Entrenamiento 18/03

USAR PLANTILLA GUARDAR CANCELAR

**Lista de ejercicios**

AÑADIR EJERCICIO

6 Partido F11

Duración Comentario

4 Finalizaciones 15

3 Centros 10

2 Partido mini-porterías 25

1 Posesiones 15

**Detalles del entreno**

Descripción

Objetivo principal

Objetivo principal

Otros comentarios

Tipo de entrenamiento

☐ Compensatorio

☐ Físico

☐ Táctico

☐ Prepartido

☐ Mantenimiento

**Figura 34. Visualización/Gestión de entrenamientos.**  
En esta interfaz se abarcan los RF6, RF7, RF8 y RF10.

#### 7.6.8. Lista de entrenos

Tenemos también la lista de entrenamientos, los cuales se podrán filtrar por autor y tipo de entrenamiento. La lista tendrá información sobre el autor del entreno, su nombre, el número de ejercicios y la fecha de creación. Dentro de ella podemos acceder a ver entrenamientos de otros usuarios o a editar y eliminar los nuestros.

En esta interfaz se cumplen los RF6 y RF12 de gestionar entrenamientos y ver histórico de entrenamientos.

Entrenamientos

DROPDOWN Select DROPDOWN Select FILTER

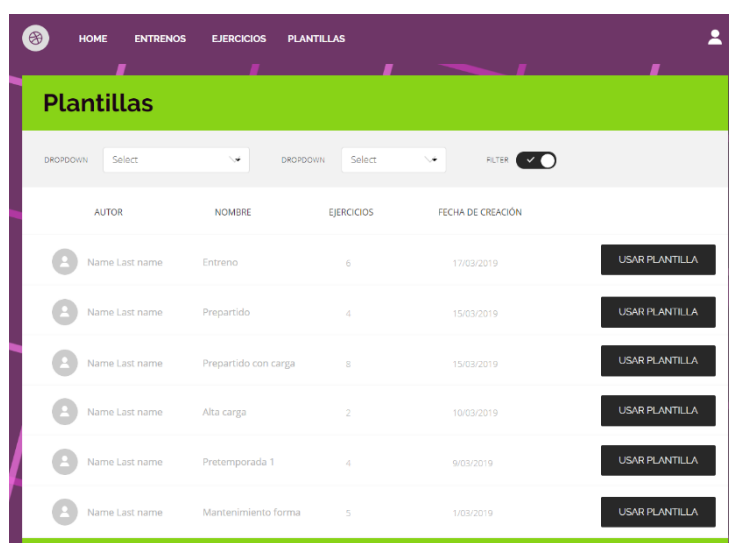
AUTOR	NOMBRE	EJERCICIOS	FECHA DE CREACIÓN	
Name Last name	Entreno	6	17/03/2019	VER/EDITAR
Name Last name	Prepartido	4	15/03/2019	VER/EDITAR
Name Last name	Prepartido con carga	8	15/03/2019	VER/EDITAR
Name Last name	Alta carga	2	10/03/2019	VER/EDITAR
Name Last name	Pretemporada 1	4	9/03/2019	VER/EDITAR
Name Last name	Mantenimiento forma	5	1/03/2019	VER/EDITAR

**Figura 35. Lista y gestión de entrenamientos.**  
En esta interfaz se abarcan los RF6 y RF12.

### 7.6.9. Lista de plantillas

Por último, tenemos la interfaz de lista de plantillas, en la cual podemos filtrar plantillas por autor y número de ejercicios. Además, en esta interfaz podemos acceder a crear un entrenamiento directamente con alguna de las plantillas encontradas gracias al botón “Usar Plantilla”. Presionando este botón, se nos cargará un nuevo entreno con el contenido de la plantilla listo para guardar el entrenamiento o guardarlo según necesitemos.

Esta interfaz cumple con los requisitos RF8, RF9 y RF10 de gestión y uso de plantillas de entrenamientos.



AUTOR	NOMBRE	EJERCICIOS	FECHA DE CREACIÓN	
Name Last name	Entreno	6	17/03/2019	USAR PLANTILLA
Name Last name	Prepartido	4	15/03/2019	USAR PLANTILLA
Name Last name	Prepartido con carga	8	15/03/2019	USAR PLANTILLA
Name Last name	Alta carga	2	10/03/2019	USAR PLANTILLA
Name Last name	Pretemporada 1	4	9/03/2019	USAR PLANTILLA
Name Last name	Mantenimiento forma	5	1/03/2019	USAR PLANTILLA

*Figura 36. Lista y gestión de plantillas.  
En esta interfaz se abarcan los RF8, RF9 y RF10.*

#### 7.6.10. Mapa de navegación

La navegación entre las pantallas de la aplicación queda definida en el mapa de navegación de la figura 37.

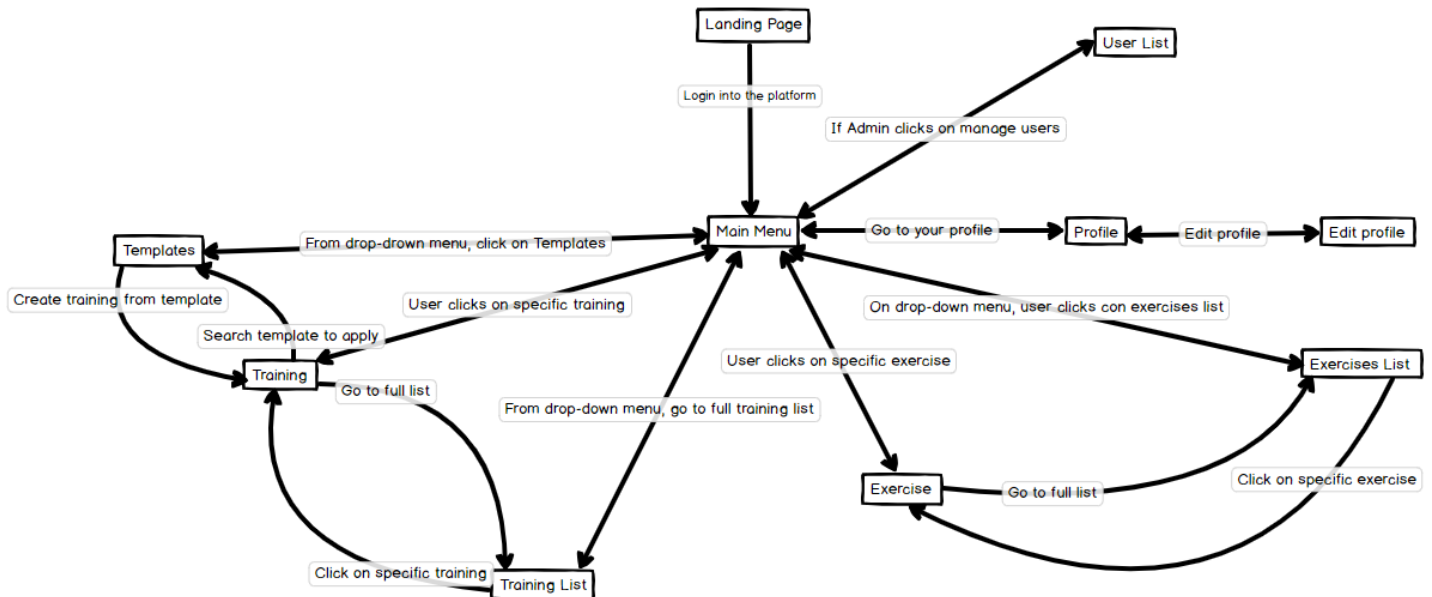


Figura 37. Mapa de navegación de la aplicación.  
(Fuente propia)

## 7.7. Guías de estilos

El objetivo de determinar una guía de estilos para nuestra aplicación es asegurarnos de que conseguimos transmitir realmente las ideas que queremos y que la aplicación entera sea coherente en su distribución de contenidos y paletas de colores usadas.

#### 7.7.1. Paleta cromática

Al entrar en nuestra aplicación, queremos transmitir una sensación de seguridad y confianza, pero con un toque rompedor y moderno. Es por ello por lo que apostamos por un coloreado minimalista con sólo dos colores principales, aunque muy diferentes entre ellos (no queremos provocar una sensación infantil con mucho colorido). Además, tendremos otros dos colores complementarios que se usarán para bordes y tipografías.

Estos colores quedan definidos en la paleta de colores de la figura 38.





Figura 38. Paleta de colores de la aplicación  
(Fuente propia)

### 7.7.2. Tipografía

Siguiendo con la misma idea que antes, con la tipografía también queremos transmitir una sensación minimalista y moderna. Para conseguir este propósito hemos buscado una fuente “Sans Serif” que huya del clásico Times New Roman y aporte un estilo fresco.

La tipografía elegida es Raleway, la cual vemos en la figura 39.

She stared  
through the  
window at the  
stars.

Figura 39. Tipografía Raleway  
(Fuente Google Fonts)

Además, se define una tabla con los tamaños para el texto en las distintas partes del sistema en función de su ubicación:

Tipo de texto	Tamaño
Títulos H1	30/36px
Títulos H2	21/27px
Títulos H3	16/21px
Títulos H4	14/18px
Texto normal	14/18px
Miniaturas	12/16px

Tabla 3. Tamaños del texto

## 7.8. Diseño de pruebas y validación

Para asegurar y mantener la calidad del producto se definen dos procesos, uno automatizado y otro manual, en los que comprobar el correcto funcionamiento del sistema después de cada lanzamiento o actualización.

El proceso automatizado se corresponde con un test de regresión [28] de la API, el cual probará toda la lógica y funcionalidades CRUD del sistema. Este test se lanzará después de cada actualización de la aplicación, sobre todo para asegurar que todo sigue funcionando según lo esperado. Para más información sobre la especificación de los test de regresión, ver Apéndice II: Test de regresión de la API.

El proceso manual de testing se corresponde con unas pruebas manuales siguiendo el journey o ruta habitual de los usuarios de la aplicación, de forma que comprobamos que todo funciona según lo esperado. Estas pruebas manuales se definen en la tabla 4 y deben ser realizadas secuencialmente.

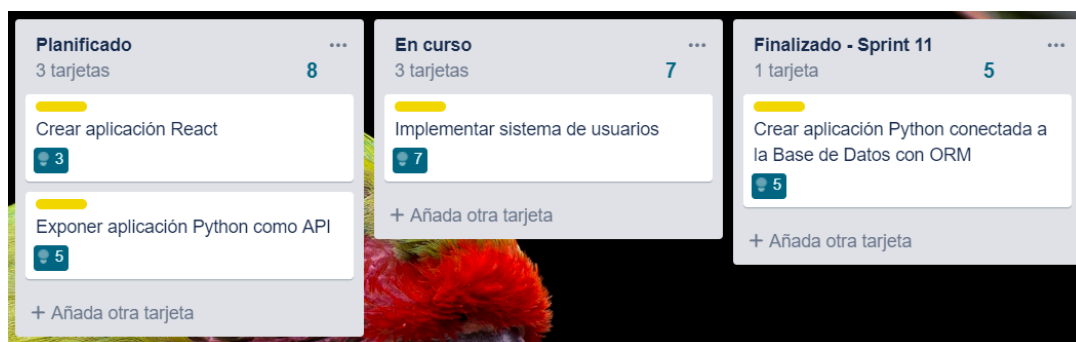
Acción	Resultado esperado
Registro en la plataforma	Usuario dado de alta y redirigido a la parte privada de la aplicación
Logout de la plataforma	Redirección a la parte pública de la aplicación
Login con datos incorrectos	Mostrar error de login incorrecto
Login correcto	Acceso a la parte privada de la aplicación
Ver todos los ejercicios	Acceso a la lista de ejercicios
Crear un nuevo ejercicio	Creación de un nuevo ejercicio
Actualizar el ejercicio creado con un nuevo dibujo y nueva información	Actualización del ejercicio
Borrar ejercicio desde la lista de ejercicios	Borrado del ejercicio del sistema
Acceder a la creación de un nuevo entrenamiento desde 0	Acceso a la interfaz de creación de entrenamientos
Añadir un ejercicio al entrenamiento y guardar el entrenamiento	Que se quede asociado el ejercicio a ese entrenamiento
Editar el entrenamiento y añadir nuevos ejercicios	Actualización del entrenamiento con la nueva información
Guardar el entreno actual como plantilla	Creación de una plantilla con la base del entrenamiento actual
Eliminar entrenamiento actual	Entrenamiento eliminado
Crear un nuevo entrenamiento a partir de una plantilla	Usar una plantilla y que al crear el entrenamiento por defecto esté rellenado con los datos de la plantilla seleccionada
Guardar el entrenamiento	Que se quede almacenado el entrenamiento en la base de datos
Ver histórico de entrenamientos	Ver una lista cronológica de los entrenamientos creados anteriormente
Ver todos los ejercicios	Acceso a la lista de ejercicios
Ver todos los templates	Acceso a la lista de templates
Editar template y añadir nuevos ejercicios	Que se quede el ejercicio actualizado acorde a los cambios
Eliminar template	Template eliminado de la base de datos

*Tabla 4. Pruebas manuales del sistema*

## 8. Implementación

Una vez terminado el diseño y especificación del proyecto, se procede con el inicio de la implementación del proyecto. Para el trabajo, desarrollaremos un pequeño prototipo con las tecnologías seleccionadas, dejando el resto de implementación de funcionalidades para trabajo futuro.

La tarea que nos ocupa ahora es la de desplegar la aplicación de forma local, usando un gestor de base de datos y desplegando las aplicaciones de Python y React. Este trabajo se corresponde con el Sprint 11 de trabajo de nuestro proyecto:



*Figura 40. Sprint 11 de trabajo  
(Fuente propia)*

En dicho Sprint, se ha generado la base de datos con el ORM de Python, por lo que la definición de tablas y relaciones se hizo mediante clases en Python que son posteriormente migradas a SQL.

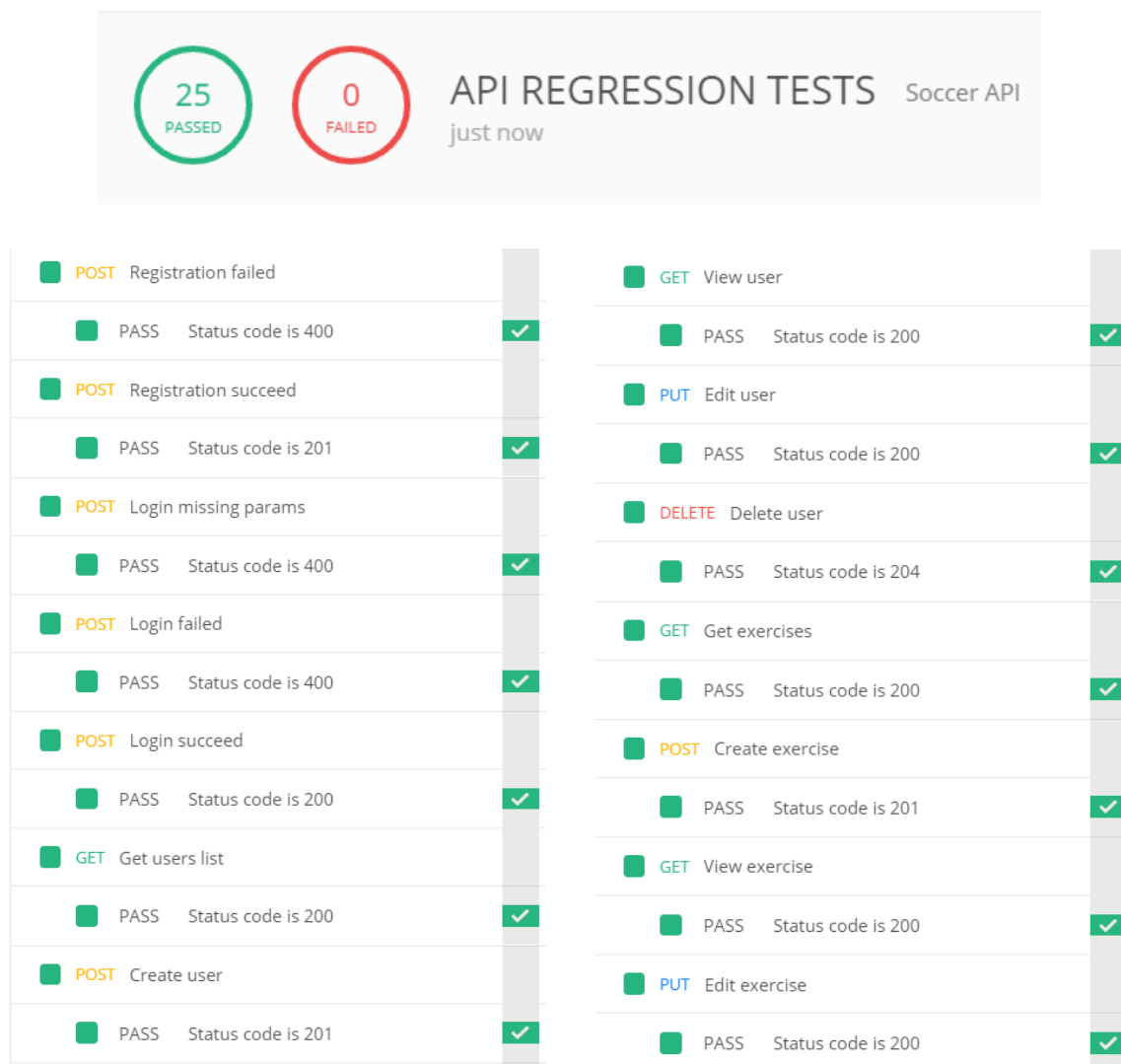
Además, gracias a la funcionalidad de Python de gestión de usuarios, se diseñó el sistema registro y login a través de línea de comandos. Posteriormente, con el uso del framework Rest API de Django se expuso todo el contenido y funcionalidad de la aplicación Python a través de una API, pudiendo hacer peticiones de datos, peticiones de registros, inicios de sesión, registros en la plataforma, etc.

Por último, como punto previo final antes de empezar a implementar lógica de negocio, se creó nuestra aplicación React, la cual se basa en React Native y es compilada y ejecutada sobre Expo-cli, lo cual nos permite correr la aplicación tanto en el navegador web como en dispositivos móviles.

## 9. Pruebas y validación

En este punto del desarrollo (primer despliegue de la aplicación), podemos ejecutar los test de regresión definidos en el Apéndice II: Test de regresión de la API. En ellos, se hace una prueba completa de la funcionalidad de la API, para asegurarnos de que, después de cada cambio, la funcionalidad de la API sigue siendo la misma y no existen cambios inesperados.

Resultados de los primeros test de regresión tras el despliegue de la API:



DELETE Delete exercise		
PASS Status code is 204	✓	
GET Get trainings		
PASS Status code is 200	✓	
POST Create training		
PASS Status code is 201	✓	
GET View training		
PASS Status code is 200	✓	
PUT Edit training		
PASS Status code is 200	✓	
DELETE Delete training		
PASS Status code is 204	✓	
GET Get templates		
PASS Status code is 200	✓	
		POST Create template
		PASS Status code is 201
		GET View template
		PASS Status code is 200
		PUT Edit template
		PASS Status code is 200
		DELETE Delete template
		PASS Status code is 204

En cada uno de los test, se comprueba que el resultado de la operación es el esperado y que el recurso es visualizado/creado/editado/borrado correctamente. En este caso, vemos que todo ha ido según lo esperado, ya que todos los asserts definidos en la colección de Postman coinciden con la respuesta de la API.

De esta manera, conforme se continúe con el desarrollo, estos test se deberán ir ejecutando tras cada cambio importante para comprobar que no estamos desestabilizando la aplicación y que todo lo realizado hasta el momento sigue funcionando correctamente (lo que nos da cierta fiabilidad a la hora de subir cambios al entorno real).

Algunos ejemplos de respuesta de nuestra API son los siguientes:

- Login en la aplicación:
  - URL: `{{API_URL}}/api/auth/`
  - Método: POST
  - Respuesta:

```
{
  "token": "d8ae2125c66c00a69658a022bac91bf8e596f9d9"
}
```

- Funcionalidad: Con el token recibido, podemos llamar a otras rutas restringidas de la aplicación, donde el token nos servirá como forma de autenticación.
- Creación de un ejercicio:
  - URL: `{{API_URL}}/api/exercises/`
  - Método: POST
  - Respuesta:
 

```
{
    "id": 8,
    "name": "MyExercise",
    "user": 3,
    "trainings": [
      1
    ]
  }
```
  - Funcionalidad: La creación del ejercicio devuelve el identificador del nuevo ejercicio, sus datos, el usuario propietario y una lista de entrenamientos en los que está incluido dicho ejercicio (recordemos que la relación entre ejercicios y entrenamientos es de muchos a muchos).

Al carecer de momento de un front-end desarrollado, la batería de pruebas manuales no se podrá realizar por ahora. Esta entrará en juego cuando se desarrollen las primeras interfaces.

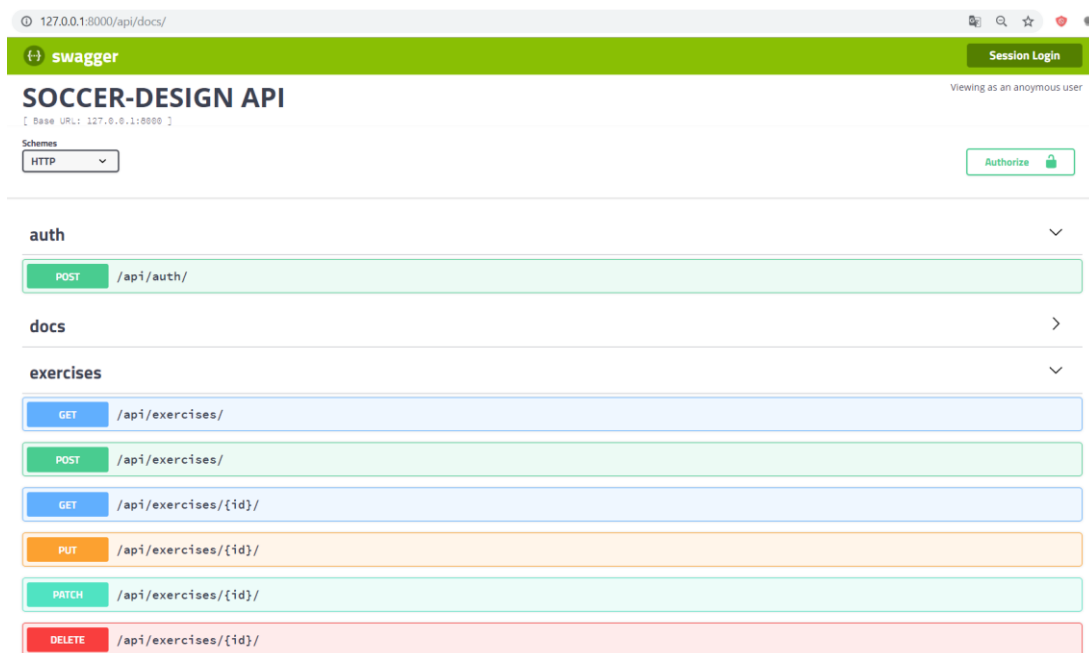
En definitiva, tenemos un buen punto de partida para empezar a desarrollar funcionalidades de la aplicación ya que tenemos una API funcional y testeable.

## 10. Resultados

Como resultado final, tenemos un proyecto suficientemente especificado y con un alcance concreto y bien definido. Además de ello, tenemos la base del proyecto construida ya que todas las partes de la aplicación (front-end, back-end y base de datos) están desplegadas, corriendo y comunicándose entre ellas correctamente. Esto último es un buen punto de partida a la hora de desarrollar ya que ya hemos superado toda la parte de configuración, solución de errores y problemas de compatibilidad y nos podemos centrar en añadir funcionalidad y valor al producto.

### 10.1. Contenido demostrable

Tenemos generada la documentación online completa de nuestra API, de la cual vemos un extracto en la figura 41. Dicha documentación nos servirá sobre todo a la hora de hacer pruebas y consultas en tiempo de desarrollo.



*Figura 41. Fragmento de la documentación de la API  
(Fuente propia)*

### 10.2. Tiempos de desarrollo

Por regla general, los tiempos planificados han sido cumplidos con éxito. Durante el desarrollo de la memoria, se iban ajustando los contenidos y profundizaciones de los apartados en función



del tiempo restante planificado para cada tarea. Eso ha hecho que se tengan unos desarrollos de apartados acordes al esfuerzo planificado inicialmente.

No podemos decir lo mismo del despliegue de la base de la aplicación. Para estas tareas del proyecto (Crear aplicación Python conectada a la Base de Datos con ORM, Implementar sistema de usuarios, Exponer aplicación Python como API y Crear aplicación React), se estimaron una serie de horas que finalmente fueron menos de las empleadas realmente en el desarrollo. En este punto, ya entraban en juego posibles bugs y problemas de las diferentes tecnologías, además de tener la necesidad de finalizar al completo la tarea que estás realizando (sin tener la posibilidad de reducir trabajo o no profundizar tanto, ya que esto generaría problemas más adelante al avanzar en funcionalidades de la aplicación). Hablamos, por ejemplo, de los siguientes contratiempos:

- Bugs o problemas de las librerías instaladas, que nos obligan a hacer workarounds para conseguir la funcionalidad deseada.
- Documentación y frameworks obsoletos, por lo que se recurre a varias fuentes para poder obtener información correcta y actualizada y teniendo que rehacer ciertos puntos del desarrollo para adaptarse a diferentes cambios (por ejemplo, se empleó una librería recomendada para desarrollar y testear la aplicación tanto en dispositivos web como en dispositivos móvil, pero finalmente resultó que esta librería no estaba mantenida y no funcionaba correctamente con las últimas versiones del framework React).
- Por último, adaptaciones más complejas de lo esperado de las funcionalidades estándar de los frameworks (por ejemplo, el sistema de usuarios por defecto de Django no se corresponde al 100% con lo que necesitamos en la aplicación en términos de información de los usuarios, formas de autenticarse, etc. Por lo que, adaptar dicho sistema de usuarios a nuestra aplicación, ha supuesto un coste mayor del esperado).

### 10.3. Contenidos de asignaturas empleados

Por último y para finalizar, es indudable que diversos contenidos de las asignaturas de la carrera son necesarios para el correcto desarrollo de un proyecto de este tipo. Por ello, pasamos a repasar las asignaturas más importantes de la carrera en cuanto a influencia dentro del proyecto:

- **Análisis y Especificación de Sistemas Multimedia:** Asignatura clave tanto para proyectos universitarios como para el mundo laboral. Realmente, como estudiante, se le da

ninguna o muy poca importancia a esta asignatura que a posteriori será el trabajo casi diario de un desarrollador.

- **Diseño de Bases de Datos Multimedia:** Tener un conocimiento amplio y analítico de las bases de datos, que cosas pueden ofrecer y como implementarlas es básico para un proyecto. Una mala especificación de base de datos puede terminar con mucha facilidad con un proyecto fallido.
- **Programación Hipermedia II:** Asignatura clave que en lo personal me hizo decantarme por el mundo del desarrollo de aplicaciones y páginas web. En esta asignatura se adquiere una práctica con JavaScript que será la base para todo el conocimiento que se adquirirá a posteriori.

En definitiva, son muchas las asignaturas que nos han aportado conocimientos valiosos para el mundo laboral y el desarrollo de este proyecto. Otras, sin embargo, tanto por parte del alumno como de la organización del grado, no se aprovechan ni se valoran como se debería.

Esto cambia una vez se tiene una visión global del grado y del sector del desarrollo de software, momento en el que se entiende y se valoran los conocimientos que se han impartido con mayor o menor éxito dentro de los cuatro años de estudios.

## 11. Conclusiones y trabajo futuro

Finalmente, se da por concluido este proyecto de forma parcialmente satisfactoria. Se han superado obstáculos como el desconocimiento y falta de práctica y agilidad a la hora de desarrollar ciertos apartados. También nos hemos enfrentado a uno de los problemas más grandes de los desarrolladores, que es el de tener la capacidad de abstraerse y definir soluciones con un respaldo y justificación lógicos. La tendencia siempre es adentrarse de lleno en el desarrollo perdiendo muchas en el camino que en muchos casos llevan al desastre.

En cambio, en este trabajo, se ha desarrollado la parte más analítica y teórica de nosotros, y en el caso de este proyecto, sirve como punto de partida para el desarrollo completo de la aplicación.

El próximo paso a seguir es continuar con el desarrollo de la aplicación según lo especificado. Una vez completo, el futuro ideal del proyecto sería integrarse con el sistema existente mencionado anteriormente de `noveldacfgestion.com`, de forma que ambos sistemas se nutran mutuamente tanto de usuarios como de información de los entrenamientos y ejercicios y que el anterior complete su funcionalidad con esta herramienta moderna, útil y completa.

A pesar de todo este trabajo futuro, el resultado del proyecto es satisfactorio. Y como hemos mencionado en el punto anterior, se considera un buen punto de partida para continuar aprendiendo y desarrollándose como profesional dentro del mundo del desarrollo del software.

## Referencias

A continuación, se listan todas las referencias mencionadas en el proyecto, aportando sus enlaces, fechas de creación y autores en los casos que corresponda:

1. Easy2Coach. <https://easy2coach.net>. 2018.
2. CampusDeportivo. <https://campusdeportivo.com>. 2006.
3. SoccerSpecific. <https://soccerspecific.com>. 2018.
4. SportEasy. <https://sporteasy.net>. 2018.
5. GesDep. <https://gestiondeportiva.com>. 2018.
6. Pamela Rodríguez. *La importancia de la Claridad y Sencillez de una interfaz de Usuario*. Disponible en: <https://sg.com.mx/revista/31/la-importancia-la-claridad-y-sencillez-una-interfaz-usuario>
7. David Arenzana. *Principios de usabilidad web de Jakob Nielsen: diseño UX*. Disponible en: <https://es.semrush.com/blog/usabilidad-web-principios-jakob-nielsen/>
8. Tomás Antón. *La usabilidad es la clave para crear aplicaciones de éxito*. Disponible en: <http://www.neolabels.com/actualidad/la-usabilidad-es-la-clave-para-crear-aplicaciones-de-exito/>
9. GeeksforGeeks. *ACID Properties in DBMS*. Disponible en: <https://www.geeksforgeeks.org/acid-properties-in-dbms/>
10. Ainhoa Lafuente. *Bases de datos relacionales vs. No relacionales: ¿qué es mejor?* Disponible en: <https://aukera.es/blog/bases-de-datos-relacionales-vs-no-relacionales/>
11. Carlos Paramio. *El concepto NoSQL, o cómo almacenar tus datos en una base de datos no relacional*. Disponible en: <https://www.genbeta.com/desarrollo/el-concepto-nosql-o-como-almacenar-tus-datos-en-una-base-de-datos-no-relacional>
12. Karim Sierra. *¿Qué es Laravel y para qué sirve?* Disponible en: <https://www.synergyweb.es/blog/laravel-desarrollo-medida/>
13. Aran Davies. *Why should you use Java for your Backend Infrastructure?* Disponible en: <https://www.devteam.space/blog/why-should-you-use-java-for-your-backend-infrastructure/>
14. Programación en Python – Nivel básico. Disponible en <https://entrenamiento-python-basico.readthedocs.io/es/latest/index.html>
15. José Manuel Roca Moncayo. *Por qué usar NodeJS*. <https://openwebinars.net/blog/por-que-usar-nodejs/>

16. CampusMVP. *Las 5 principales ventajas de usar Angular para crear aplicaciones web*. Disponible en: <https://www.campusmvp.es/recursos/post/las-5-principales-ventajas-de-usar-angular-para-crear-aplicaciones-web.aspx>
17. Miguel Ángel Álvarez. *Qué es React. Por qué usar React*. Disponible en: <https://desarrolloweb.com/articulos/que-es-react-motivos-uso.html>
18. Carlos Azaustre. *Qué es lo que me gusta de Vue.js*. Disponible en: <https://carlosazaustre.es/que-es-lo-que-me-gusta-de-vue-js/>
19. Andrei Neagoie. *React vs Angular vs Vue: Who wins in 2019?* Disponible en: <https://medium.com/zerotomastery/react-vs-angular-vs-vue-who-wins-in-2019-5d9acd0843e8>
20. Jonny Kalambay. *Your First Hybrid App in Minutes – React Native on the Web*. Disponible en: <https://medium.com/@jonnykalambay/your-first-hybrid-app-in-15-minutes-react-native-on-the-web-2cc2646051e>
21. Tic Portal. *Waterfall: metodología para el desarrollo secuencial de tareas*. Disponible en: <https://www.ticportal.es/glosario-tic/waterfall-metodologia-desarrollo-secuencial>
22. Arturo León. *Scrum para uno*. Disponible en: <https://medium.com/medium-en-espanol/scrum-para-uno-a895de6010a4>
23. Ignacio Pérez. *¿En qué consiste la vulnerabilidad Cross Site Request Forgery (CSRF)?* Disponible en: <https://www.welivesecurity.com/la-es/2015/04/21/vulnerabilidad-cross-site-request-forgery-csrf/>
24. HostingPedia. *MySQL*. Disponible en: <https://hostingpedia.net/mysql.html>
25. Trellat. *MyISAM vs InnoDB: ventajas y desventajas*. Disponible en: <https://trellat.es/myisam-vs-innodb-ventajas-desventajas/>
26. Facebook. *React Native*. Disponible en: <https://facebook.github.io/react-native/>
27. Expo CLI. Disponible en: <https://docs.expo.io/versions/latest/workflow/expo-cli/>
28. Ana M. del Carmen García Oterino. *Pruebas de regresión. O cómo evitar que los fallos que se habían arreglado en el software vuelvan a aparecer*. Disponible en: <https://www.javiergarzas.com/2014/06/pruebas-de-regresion.html>

## Apéndice I: Especificación de la API

Nuestra API es la encargada de comunicar nuestra aplicación React con la base de datos. Por lo general, cada objeto tiene permitidas las operaciones CRUD además de una lista de registros. Sin embargo, existen algunas rutas especiales como las de la documentación o las de inicio de sesión de usuarios. Las rutas y métodos disponibles en la API quedan definidos en la siguiente lista:

- **/AUTH/** Ruta para manejar la autenticación de usuarios.
  - POST /api/auth/
- **/DOCS/** Ruta donde obtener una documentación completa de nuestra API.
  - GET /api/docs/
- **/EXERCISES/** Ruta para realizar toda la gestión y visualización de los ejercicios del sistema.
  - GET /api/exercises/
  - POST /api/exercises/
  - GET /api/exercises/{id}/
  - PUT /api/exercises/{id}/
  - PATCH /api/exercises/{id}/
  - DELETE /api/exercises/{id}/
- **/LOGS/** Ruta donde manejar la administración de logs del sistema. No tiene funcionalidad por sí misma, pero puede ser útil para registrar errores y depurar nuestra aplicación.
  - GET /api/logs/
  - POST /api/logs/
  - GET /api/logs/{id}/
  - PUT /api/logs/{id}/
  - PATCH /api/logs/{id}/
  - DELETE /api/logs/{id}/
- **/TEMPLATES/** Las plantillas de entrenamientos también tendrán su propia ruta para poder ser gestionadas, que incluirá las operaciones CRUD.
  - GET /api/templates/
  - POST /api/templates/
  - GET /api/templates/{id}/

- PUT /api/templates/{id}/
- PATCH /api/templates/{id}/
- DELETE /api/templates/{id}/
- **/TRAININGS/** Al igual que las plantillas y los ejercicios, los entrenamientos también tienen sus métodos CRUD y un método de visualización de una lista de ejercicios.
  - GET /api/trainings/
  - POST /api/trainings/
  - GET /api/trainings/{id}/
  - PUT /api/trainings/{id}/
  - PATCH /api/trainings/{id}/
  - DELETE /api/trainings/{id}/
- **/USERS/** Por último, tenemos la ruta de gestión de usuarios, que será utilizada básicamente por los administradores. Algunas rutas (como la de edición de usuarios) serán accesibles por los usuarios estándar siempre que sean para hacer cambios en sus propios perfiles.
  - GET /api/users/
  - POST /api/users/
  - GET /api/users/{id}/
  - PUT /api/users/{id}/
  - PATCH /api/users/{id}/
  - DELETE /api/users/{id}/

## Apéndice II: Test de regresión de la API

El test de regresión de la API define un test secuencial de los métodos funcionales de la API, de forma que cada vez que lo ejecutemos podremos comprobar si cada uno de los métodos sigue funcionando según lo esperado.

Otra ventaja de este test es que, de cara a despliegues de la aplicación en diferentes entornos, el scope del test es fácilmente intercambiable para apuntar a una URL u otra y poder testear todos los entornos en los que tenemos desplegada la aplicación. Para nuestra API se definen las siguientes pruebas divididas por funcionalidad:

- La primera parte es de registros y autenticaciones, con los que probaremos distintas entradas de datos para obtener respuestas correctas o fallidas:

```
graph TD; A[POST Registration failed] --> B[POST Registration succeed]; B --> C[POST Login missing params]; C --> D[POST Login failed]; D --> E[POST Login succeed];
```

POST Registration failed  
POST Registration succeed  
POST Login missing params  
POST Login failed  
POST Login succeed

*Figura 42. Testeo de registros y autenticaciones  
(Fuente propia)*

- Por otra parte, probaremos toda la gestión de usuarios que será realizada principalmente por usuarios administradores:

```
graph TD; A[GET Get users list] --> B[POST Create user]; B --> C[GET View user]; C --> D[PUT Edit user]; D --> E[DEL Delete user];
```

GET Get users list  
POST Create user  
GET View user  
PUT Edit user  
DEL Delete user

*Figura 43. Testeo de la gestión de usuarios  
(Fuente propia)*

- Entrando ya en la lógica de negocio, se definen las siguientes pruebas para testear el correcto funcionamiento de los métodos de los ejercicios:



GET	Get exercises
POST	Create exercise
GET	View exercise
PUT	Edit exercise
DEL	Delete exercise

*Figura 44. Testeo de los métodos de ejercicios  
(Fuente propia)*

- Además de los ejercicios, tenemos los tests de los entrenamientos y sus operaciones:

GET	Get trainings
POST	Create training
GET	View training
PUT	Edit training
DEL	Delete training

*Figura 45. Testeo de los entrenamientos y sus métodos  
(Fuente propia)*

- Por último, se testea la funcionalidad de las plantillas de entrenamientos:

GET	Get templates
POST	Create template
GET	View template
PUT	Edit template
DEL	Delete template

*Figura 46. Testeo de las plantillas de entrenamientos y sus métodos  
(Fuente propia)*